



Gentoo Linux

Instrukcja instalacji i konfiguracji systemu

Spis treści

A. Instalacja Gentoo.....	3
1. O instalacji Gentoo Linux.....	3
2. Wybór medium instalacyjnego.....	4
3. Konfigurowanie sieci.....	9
4. Przygotowywanie dysków.....	15
5. Wypakowywanie plików instalacyjnych Gentoo.....	21
6. Instalowanie systemu podstawowego.....	26
7. Konfigurowanie jądra.....	30
8. Konfigurowanie systemu.....	36
9. Instalowanie narzędzi systemowych.....	40
10. Konfiguracja bootloadera.....	43
11. Zakończenie instalacji Gentoo.....	49
12. I co dalej?.....	49
B. Praca z Gentoo.....	50
1. Wprowadzenie do Portage.....	50
2. Flagi USE.....	58
3. Funkcje Portage.....	62
4. Skrypty startowe.....	65
5. Zmienne środowiskowe.....	72
C. Praca z Portage.....	75
1. Pliki i katalogi.....	75
2. Konfigurowanie Portage.....	77
3. Mieszanie różnych gałęzi Portage.....	79
4. Dodatkowe narzędzia Portage.....	81
5. Pozostawiając oficjalne drzewo Portage.....	83
D. Konfiguracja sieci w Gentoo.....	84
1. Wprowadzenie.....	84
2. Zaawansowana konfiguracja.....	85
3. Modularna praca w sieci.....	88
4. Połączenia bezprzewodowe.....	93
5. Dodawanie możliwości.....	98
6. Zarządzanie siecią.....	100
E. Dokumentacja dla Gentoo na stacjach roboczych.....	101
1. Instalacja i konfiguracja serwera X.....	101
2. Instalacja i konfiguracja kart nVidia.....	106
3. Instalacja i konfiguracja kart ATI.....	112
4. Instalacja i konfiguracja KDE.....	114
5. Rozdzielone ebuildy w KDE.....	121
6. Instalacja i konfiguracja Gnome.....	124
7. Instalacja i konfiguracja Fluxbox.....	126
8. Przewodnik po sprzętowej akceleracji 3D.....	132
9. Kodowanie UTF-8 w Gentoo.....	138
10. Lokalizacja Gentoo Linux.....	147
11. Konfiguracja sterowników ALSA w Gentoo.....	152
12. Java w Gentoo.....	162
13. Instalacja i konfiguracja CUPS.....	168
F. Administrowanie Gentoo.....	175
1. Bezpieczeństwo systemu.....	175
2. Uszczelnianie systemu.....	178
3. Logowanie.....	180
4. Montowanie partycji.....	184
5. Limitowanie grup i użytkowników.....	185
6. Prawa dostępu do plików.....	188
7. PAM.....	190
8. TCP Wrappers.....	191
9. Bezpieczeństwo jądra.....	192
10. Zabezpieczanie usług.....	194
11. Chrootowanie i serwery wirtualne.....	201
12. Firewall.....	202
13. Wykrywanie włamań.....	213
14. Aktualizacje.....	218
15. Prelink w Gentoo.....	220
16. Instalacja i konfiguracja demona CRON.....	223
17. Aktualizacja jądra systemu.....	229
18. Konfiguracja jądra systemu Linuks.....	233
19. Praca z distcc w Gentoo.....	238
20. Konfiguracja routera.....	242
21. Optymalizacja komplikacji.....	252

A. Instalacja Gentoo

1. O instalacji Gentoo Linux

1.a. Wprowadzenie

Po pierwsze witamy w Gentoo. Wkraczasz w świat szerokich możliwości i dużej wydajności. Możliwość wyboru to podstawowa zaleta naszej dystrybucji. Podczas instalacji można zdecydować jak dużą część systemu pragnie się zbudować samodzielnie, który program logujący ma pracować w systemie itd.

Gentoo to szybka i nowoczesna dystrybucja. Do jej głównych zalet należą przejrzystość i elastyczność. Tworzymy je jako wolne oprogramowanie i staramy się nie ukrywać niczego przed użytkownikiem. Portage, czyli nasz system zarządzania pakietami napisaliśmy w Pythonie, dzięki czemu można z łatwością przeglądać i modyfikować jego kod tak, aby dostosować go do swoich potrzeb. Gentoo jest oparte głównie na pakietach źródłowych, ale posiada również wsparcie dla pakietów prekompilowanych. Cała konfiguracja odbywa się za pomocą zwyczajnych plików tekstowych. Podsumowując: Gentoo to pełna otwartość.

Niezwykle istotne jest zrozumienie, czemu możliwość wyboru jest aż tak ważna. Nie próbujemy zmuszać użytkowników do robienia czegoś, czego nie chcą. Jeśli uważasz, że w jakimś przypadku powinniśmy, [powiadom nas](#) o tym.

Jak przebiega instalacja?

Proces instalacji Gentoo można podzielić na 10 etapów, opisanych odpowiednio w rozdziałach 2 - 11. Każdy z nich kończy się w określonym momencie:

- Po ukończeniu etapu pierwszego użytkownik znajduje się wewnętrz w pełni skonfigurowanego i przygotowanego do pracy środowiska instalacyjnego.
- Po ukończeniu etapu drugiego możemy korzystać z właśnie skonfigurowanego łącza internetowego.
- Po ukończeniu etapu trzeciego dyski i partie w komputerze są gotowe do zainstalowania Gentoo.
- Po ukończeniu etapu czwartego środowisko instalacyjne jest w pełni przygotowane i można zalogować się do systemu.
- Po ukończeniu etapu piątego są zainstalowane wszystkie podstawowe pakiety.
- Po ukończeniu etapu szóstego jądro Linuksa jest przygotowane do pracy.
- Po ukończeniu etapu siódmego mamy naniesione odpowiednie poprawki na większość plików konfiguracyjnych.
- Po ukończeniu etapu ósmego mamy zainstalowane niezbędne narzędzia systemowe.
- Po ukończeniu etapu dziewiątego mamy zainstalowany i skonfigurowany bootloader. Możemy też zalogować się do świeżo zainstalowanego systemu.
- Po ukończeniu etapu dziesiątego proces instalacji został zakończony i można przystąpić do odkrywania ogromnych możliwości Gentoo.

Za każdym razem gdy użytkownik będzie zmuszony do wybrania jednej z kilku opcji postaramy się jak najlepiej przedstawić wady i zalety każdego z rozwiązań. Następnie będziemy kontynuować omawianie procesu instalacji opisując kolejno wybór domyślny, a następnie wszystkie alternatywne możliwości. Domyślne opcje nie są tymi zalecanymi, po prostu przy pisaniu dokumentacji zakładamy, że wybierze je większość użytkowników.

Część dokumentacji jest opcjonalna. Zwykle konieczność korzystania z niej wynika z wcześniejszych wyborów użytkownika i jeśli nie dotyczy naszego przypadku spokojnie możemy ją pominąć.

Co mamy do wyboru?

Gentoo można zainstalować na wiele różnych sposobów. Najczęściej wybierana metoda to ta przy użyciu jednej z naszych płyt instalacyjnych. Istnieje również możliwość przeprowadzenia tego procesu poprzez już zainstalowaną dystrybucję, inną uruchamialną płytę (np. Knoppix), środowisko uruchamiane z sieci (netmount), czy dyskietkę ratunkową.

W Podręczniku omawiamy instalację przy użyciu płyt Gentoo LiveCD oraz, w pewnych przypadkach, rozruchu z sieci (netboot). Generalnie zakładamy, że użytkownik zamierza zainstalować najnowsze dostępne wersje pakietów. Instalację bez dostępu do Internetu omawiamy w Podręcznikach Gentoo 2006.1.

W przypadku instalacji przy pomocy GRP (Gentoo Reference Platform - kolekcja prekompilowanych pakietów służących do błyskawicznego instalowania Gentoo) trzeba skorzystać z instrukcji w Podręcznikach Gentoo 2006.1.

Przewodnik po alternatywnych metodach instalacji to dobre źródło informacji na temat mniej konwencjonalnych sposobów instalowania Gentoo. Ponadto warto zapoznać się z dokumentem zawierającym przydatne rady dotyczące instalacji Gentoo. Zaawansowani użytkownicy, którzy uważają, że w Podręczniku proces instalacji jest omówiony zbyt rozwlekłe powinni skorzystać z dokumentu opisującego wszystkie czynności w mocno skróconej formie, który znajduje się w naszych zasobach dokumentacji.

Mogliwości wyboru nie kończą się na medium instalacyjnym. Możliwe jest skompilowanie całego systemu od podstaw lub wykorzystanie do jego budowy prekompilowanych pakietów, co zaoszczędzi mnóstwo czasu. Oczywiście są także rozwiązania mieszane, dzięki którym nie trzeba kompilować wszystkiego i można zacząć od częściowo przygotowanego systemu.

Problemy?

Jeśli w czasie instalacji pojawi się jakiś problem (lub wystąpią błędy w dokumentacji) zachęcamy do odwiedzenia naszej bugzilli i sprawdzenia czy został on już zgłoszony. Jeśli jeszcze o nim nie wiemy prosimy o wypełnienie i wysłanie odpowiedniego formularza. Nie należy się bać developerów, do których zostanie przypisany raport, zwykle nie gryzą.

Pomimo, że spora część Podręcznika jest wspólna dla wszystkich architektur istnieją w nim również odnośniki do poszczególnych z nich. Staramy się ograniczać to zjawisko do minimum, aby uniknąć dezorientowania czytelników.

Jeśli nie wiadomo czy kłopot leży po stronie systemu (pewne rzeczy mogą nie być dostatecznie przetestowane) czy po stronie użytkownika (czasami problem może wynikać z nieuważnego czytania opisu) warto odwiedzić kanał #gentoo na sieci irc.freenode.net. Zapraszamy tam wszystkich użytkowników.

Odpowiedzi na wiele pytań związanych z Gentoo znajdują się w naszym FAQ. Warto również przejrzeć FAQ na naszym forum. Jeśli odpowiedzi na pytanie nie ma w żadnym z nich zawsze można zapytać maniaków przesiadujących na kanale #gentoo (w sieci freenode), zwykle są dobrze poinformowani.

2. Wybór medium instalacyjnego

2.a. Wymagania sprzętowe

Wprowadzenie

Zanim zaczniemy musimy poznać wymagania sprzętowe jakie powinien spełniać komputer, aby można było na nim zainstalować Gentoo.

Wymagania sprzętowe

	Płyta minimalna	LiveCD
Procesor	i486 lub nowszy	i686 lub nowszy
Pamięć	64 MB	128 MB
Wolne miejsce na dysku	1.5 GB (bez miejsca potrzebnego na partycję wymiany)	
Miejsce na partycję wymiany		Co najmniej 256 MB

2.b. Płyty instalacyjne Gentoo

Wprowadzenie

Dzięki płytom instalacyjnym można uruchomić podstawowe środowisko służące do instalacji całego systemu Gentoo na dysk. W czasie uruchamiania systemu z płyty zostanie wykryty sprzęt dostępny w komputerze, a Gentoo automatycznie załaduje odpowiednie dla niego sterowniki. Płyty te są tworzone i rozwijane przez deweloperów Gentoo.

Wszystkie płyty instalacyjne pozwalają na uruchomienie środowiska instalacyjnego, skonfigurowanie ustawień sieciowych, utworzenie i aktywowanie partycji oraz rozpoczęcie instalowania Gentoo z użyciem Internetu. Są idealnym środkiem do zbudowania naszej dystrybucji, jeśli zakłada się instalowanie przy użyciu najnowszych pakietów pobieranych z sieci.

Ważne: Opis instalacji Gentoo bez dostępu do Internetu znajduje się na stronach Podręcznika Gentoo 2006.1.

Są następujące rodzaje płyt instalacyjnych:

- "Gentoo Minimal Installation CD" - mała, uruchamialna płyta instalacyjna zawierająca wszystko co niezbędne do uruchomienia środowiska, z którego można następnie przystąpić do instalowania Gentoo.
- "Gentoo Installer LiveCD" - płyta zawierająca wszystkie składniki konieczne do instalacji systemu Gentoo. Na płycie tej znajduje się graficzne środowisko pracy oraz dwie wersje instalatora, z których jedną można uruchomić w konsoli.

Aby ułatwić wybór odpowiedniej płyty instalacyjnej omówimy plusy i minusy każdej z nich.

Gentoo Minimal Installation CD

Obraz tej płyty ma nazwę install-x86-minimal-2006.1.iso i zajmuje jedynie 49 MB miejsca. Można go użyć do zainstalowania Gentoo wyłącznie jeśli posiada się połączenie z Internetem.

Minimal Installation CD

Plusy i minusy

+

Mała ilość danych do pobrania.

-

Nie zawiera żadnych pliku "stage3", obrazów drzewa Portage, prekomplikowanych pakietów i w związku z tym nie nadaje się do instalacji bez dostępu do sieci.

Gentoo Installer LiveCD

Obraz tej płyty ma nazwę livecd-i686-installer-2006.1.iso i zajmuje około 697 MB miejsca. Można użyć go do zainstalowania Gentoo nawet bez dostępu do sieci.

Installer LiveCD Plusy i minusy

- | | |
|---|----------------------------------------------------------------------------------------------------|
| + | Zawiera wszystko co jest potrzebne do zainstalowania Gentoo, nawet bez dostępu do Internetu |
| - | Duża ilość danych do pobrania |

Archiwum stage3

Plik ten to archiwum zawierające podstawowy system Gentoo, z którego będzie można kontynuować instalację w sposób opisany w tym Podręczniku. W przeszłości znajdowały się tu instrukcje dotyczące instalacji z trzech różnych plików tego typu. Wciąż udostępniamy archiwia stage1 i stage2, ale w oficjalnym opisie instalacji prezentujemy jedynie czynności związane z użyciem stage3. Informacje dotyczące instalacji za pomocą stage1 i stage2 znajdują się w Gentoo FAQ w podrozdziale Jak zainstalować Gentoo za pomocą stage1 lub stage2?

Pliki stage3 można pobrać z katalogu releases/x86/2006.1/stages/ na każdym z naszych Serwerów lustrzanych. Pliki te nie znajdują się na płytach LiveCD.

2.c. Pobieranie, nagrywanie i uruchamianie płyt instalacyjnych Gentoo

Pobieranie i nagrywanie płyt instalacyjnych

Po pierwsze musimy pobrać jeden z omawianych wcześniej obrazów płyt, żeby jednak to zrobić trzeba wiedzieć gdzie ich szukać.

Wszystkie obrazy płyt instalacyjnych znajdują się na naszych serwerach lustrzanych w katalogu releases/x86/2006.1/installcd/.

Wewnątrz tego katalogu znajduje się zbiór plików ISO. Są to pełne i gotowe do nagrania obrazy płyt CD.

Aby zweryfikować poprawność pobranych plików ISO należy porównać ich sumy MD5 z tymi znajdującymi się na naszym serwerze lustrzanym (np. w pliku o nazwie install-x86-minimal-2006.1.iso.DIGESTS). Sumy MD5 dla pobranych plików można wygenerować przy pomocy narzędzia md5sum dla Linuksa, lub jego odpowiednika dla Windows.

Innym sposobem sprawdzania poprawności pobranych plików jest weryfikacja ich podpisów przy pomocy GnuPG. Sygnatury poprawnych plików znajdują się w plikach z rozszerzeniem .asc. Najpierw pobieramy plik sygnatury, a następnie pozyskujemy klucz publiczny:

Listing 1: Pozyskiwanie klucza publicznego

```
$ gpg --keyserver subkeys.pgp.net --recv-keys 17072058
```

Następnie weryfikujemy podpis.

Listing 2: Verify the cryptographic signature

```
$ gpg --verify <signature file> <downloaded iso>
```

Pobrane pliki ISO należy nagrywać w trybie RAW. To jak się go włącza zależy od programu, którego używamy. W Podręczniku opiszemy nagrywanie za pomocą programów cdrecord i K3B. Więcej informacji można znaleźć w dokumencie [Gentoo FAQ](#).

- Jeśli chodzi o cdrecord to wystarczy wpisać polecenie cdrecord dev=/dev/hdc <pobrany plik iso>. Zamiast /dev/hdc należy podać odpowiednią ścieżkę do urządzenia CD-RW.
- W K3B trzeba wybierać z menu kolejno zakładki Tools > CD > Burn Image. Następnie wskazujemy nasz plik ISO i klikamy w przycisk Start.

Uruchamianie płyty instalacyjnej

Po nagraniu płyty instalacyjnej przychodzi pora na uruchomienie z niej systemu. Upewniamy się, że w napędach nie ma żadnych innych płyt i ponownie uruchamiamy komputer. Wchodzimy do BIOS-u, zwykle robi się to przy pomocy klawiszy DEL, F1 lub ESC - zależnie od producenta i modelu płyty głównej. Wszystko co musimy tam zmienić to kolejność w jakiej będą skanowane nasze napędy w poszukiwaniu aktywnego systemu operacyjnego. Musimy skonfigurować to tak, by CD-ROM był pierwszym takim napędem. Jeśli nie zmienimy tej opcji komputer uruchomi się z dysku, ignorując naszą płytę instalacyjną.

Teraz możemy umieścić płytę w napędzie i po raz kolejny zrestartować komputer. Po pewnym czasie powinien pojawić się znak zachęty (prompt). Mamy tu dwie możliwości. Możemy od razu nacisnąć enter i rozpoczęć proces uruchamiania Gentoo z płyty instalacyjnej ze standardowymi opcjami lub dodać kilka własnych ustawień. Dokonuje się tego wpisując nazwę wybranego jądra i opcji z jakimi chcemy je uruchomić, a następnie naciskając enter.

Wybranego jądra? Tak, na płytach instalacyjnych znajduje się kilka jąder Linuksa. Domyślne i najbardziej uniwersalne z nich ma nazwę gentoo. Pozostałe jądra są skonfigurowane pod specyficzny sprzęt, oraz posiadają warianty bez sterowników graficznych (-nofb).

Oto krótkie omówienie wszystkich dostępnych jąder:

Jądro	Opis
gentoo	Standardowe jądro serii 2.6 ze wsparciem dla komputerów wieloprocesorowych
gentoo-nofb	Jądro gentoo pozbawione sterowników framebuffera
memtest86	Program do testowania pamięci RAM

Można również wybrać opcje z jakimi zostanie uruchomione jądro, kontrolując one jego określone ustawienia. Poniższa tabela przedstawia ich listę, taką samą otrzymamy naciskając przycisk F2.

Listing 3: Opcjonalne ustawienia jądra na płyce instalacyjnej

```
- agpgart      ładuje agpgart (używane w przypadku problemów z grafiką, zwłaszcza zawieszania się komputera)
- acpi=on     włącza wsparcie dla ACPI
- ide=nodma   wymusza wyłączenie DMA na psujących się napędach IDE
- doscsi      skanuje w poszukiwaniu urządzeń scsi (może nie działać prawidłowo z niektórymi kartami ethernet)
- dopcmcia    uruchamia obsługę cdromów PCMCIA
- nofirewire  wyłącza moduły firewire (dla cdromów firewire, etc)
- nokeymap    uniemożliwia wybór mapowania klawiatury innego niż US
- docache     cache'uje cały proces uruchamiania cd w RAM-ie, umożliwia to odmontowanie cdromu podczas instalacji
- nodetect   uniemożliwia uruchomienie hwsetup/kudzu i hotplug
- nousb       wyłącza ładowanie modułu usb z initrd, wyłącza też hotplug
- nodhcp      dhcp nie wystartuje automatycznie jeśli zostanie wykryty NIC
- nohotplug   wyłącza ładowanie urządzeń hotplug
- noapic      wyłącza apic (użyj w przypadku problemów ze sprzętem nics, scsi, etc)
- hdx=stroke  pozwala na partycjonowanie całego dysku, nawet wtedy, gdy BIOS nie obsługuje dużych dysków
```

Pora na uruchomienie systemu z płyty. Wybieramy jądro (jeśli domyślne gentoo nas nie zadowala) oraz opcje z jakimi ma zostać ono uruchomione. Jako przykład podamy linię uruchamiającą jądro gentoo z opcją dopcmcia.

Listing 4: Uruchamianie systemu z płyty instalacyjnej

```
boot: gentoo dopcmcia
```

Jeśli instalujemy Gentoo w systemie, w którym mamy klawiaturę inną niż US musimy wcisnąć ALT+F1, aby przejść do trybu potwierdzania kolejnych czynności, a następnie postępować zgodnie ze wskazówkami na ekranie. Jeśli nie wybierzemy nowego mapowania w ciągu 10 sekund, zostanie załadowane to domyślne, czyli amerykańskie. Jak tylko skończy się proces wczytywania systemu zostaniemy automatycznie zalogowani do "Live" Gentoo Linux jako "root", nazywany też czasem superużytkownikiem. Na bieżącej konsoli powinien pojawić się znak zachęty ("#") roota. Konsole zmieniamy kombinacjami klawiszy Alt-F2, Alt-F3 i Alt-F4. Do konsoli, którą widzieliśmy na początku wracamy naciskając Alt-F1.

Konfigurowanie dodatkowego sprzętu

W czasie uruchamiania systemu spróbuje wykryć sprzęt i załadować odpowiednie sterowniki. Zazwyczaj czyni to prawidłowo, ale czasami mogą zdarzyć się problemy i nie wszystkie moduły zostaną aktywowane. Gdy zawiedzie skanowanie PCI musimy ręcznie załadować odpowiednie moduły.

W poniższym przykładzie spróbujemy załadować moduł 8139too (obsługujący całą serię urządzeń sieciowych):

Listing 5: Ładowanie modułów jądra

```
# modprobe 8139too
```

Obsługę PCMCIA włącza się ładując odpowiedni skrypt startowy:

Listing 6: Uruchamianie skryptu startowego PCMCIA

```
# /etc/init.d/pcmcia start
```

Opcjonalnie: Poprawianie wydajności twardego dysku

Zaawansowanych użytkowników na pewno zainteresuje możliwość zwiększenia wydajności twardych dysków IDE za pomocą programu hdparm. Obecną wydajność można przetestować za pomocą parametrów -T (kilokrotne wykonanie polecenia zwiększa precyzję pomiaru):

Listing 7: Testowanie wydajności twardego dysku

```
# hdparm -T /dev/hda
```

Aby poprawić wydajność można wykorzystać któryś z poniższych przykładów (lub eksperymentować samodzielnie). Oczywiście musimy zastąpić /dev/hda ścieżką do naszego dysku.

Listing 8: Poprawianie wydajności dysku

(Aktywowanie DMA)

```
# hdparm -d 1 /dev/hda
```

(Aktywowanie zestawu bezpiecznych opcji poprawiających wydajność)

```
# hdparm -d 1 -A 1 -m 16 -u 1 -a 64 /dev/hda
```

Opcjonalnie: Konta użytkowników

Jeśli planujemy umożliwienie innym osobom dostępu do środowiska instalacyjnego lub zamierzamy korzystać z irssi nie uruchomionego z przywilejami roota musimy stworzyć dodatkowe konta.

Najpierw jednak należy zmienić hasło roota. Dokonuje się tego przy pomocy polecenia passwd:

Listing 9: Zmiana hasła roota

```
# passwd  
New password: (Podajemy nowe hasło)  
Re-enter password: (Potwierdzamy nowe hasło)
```

Aby stworzyć konto użytkownika musimy najpierw podać jego parametry, a następnie ustawić hasło. Skorzystamy przy tym z poleceń useradd oraz passwd. W przykładzie stworzymy użytkownika o nazwie "rane".

Listing 10: Tworzenie konta użytkownika

```
New password: (Podajemy hasło)  
Re-enter password: (Potwierdzamy hasło)
```

Aby przełączyć się z konta roota na nowo utworzone konto użytkownika korzystamy z polecenia su:

Listing 11: Przelaczanie użytkownika

```
# su - rane
```

Opcjonalnie: Dostęp do dokumentacji podczas instalowania Gentoo

Jeśli zamierzamy podczas instalacji korzystać z Podręcznika Gentoo (obojętnie czy nagranego na CD czy znajdującego się w Internecie) powinniśmy dodać dla tych celów konto zwykłego użytkownika, tak jak opisaliśmy to przed chwilą, a następnie przejść przy pomocy kombinacji klawiszy Alt-F2 na nowy terminal i tam się zalogować.

Do przeglądania dokumentacji nagranej na CD służy program links:

Listing 12: Przeglądanie dokumentacji na CD

```
# links /mnt/cdrom/docs/html/index.html
```

Najnowszą i najlepszą dostępną wersją Podręcznika Gentoo jest ta znajdująca się na naszej stronie internetowej. Polecamy korzystanie właśnie z tej wersji. Podobnie jak w przypadku dokumentacji nagranej na CD można użyć do tego programu links, pod warunkiem oczywiście, że mamy już skonfigurowane i działające połączenie z Internetem.

Listing 13: Przeglądanie dokumentacji w Internecie

```
# links http://www.gentoo.org/doc/en/handbook/handbook-x86.xml
```

Na pierwszy terminal powracamy przy pomocy kombinacji klawiszy Alt-F1.

Opcjonalnie: Uruchamianie demona SSH

Aby umożliwić innym osobom dostęp do naszego komputera podczas instalacji (by mogły nam pomóc w konfigurowaniu Gentoo lub nawet przeprowadziły cały proces za nas) musimy dodać im odpowiednie konta użytkowników lub nawet podać hasło roota (nie należy tego robić jeśli nie jest to osoba, której ufa się całkowicie).

Demona SSH uruchamia się następującym poleceniem:

Listing 14: Uruchamianie demona SSH

```
# /etc/init.d/sshd start
```

Korzystanie z sshd jest możliwe tylko wtedy, gdy komputer jest połączony z Internetem. Połączenie nawiążemy dzięki wskazówkom spisanym, w rozdziale zatytułowanym konfiguracja sieci.

3. Konfigurowanie sieci

3.a. Automatyczne wykrywanie sieci

Może po prostu to już działa?

Jeżeli komputer jest podłączony do sieci Ethernet z serwerem, DHCP jest bardzo prawdopodobne, że połączenie zostało skonfigurowane automatycznie. Dzięki temu od razu można skorzystać z wielu narzędzi sieciowych dostępnych na płycie instalacyjnej, takich jak ssh, scp, ping, irssi, wget czy links.

Jeśli sieć jest skonfigurowana prawidłowo to polecenie /sbin/ifconfig powinno wyświetlić oprócz lo także inne urządzenia, na przykład eth0:

Listing 1: Wynik /sbin/ifconfig przy poprawnej konfiguracji sieci

```
# /sbin/ifconfig
(...)
eth0      Link encap:Ethernet HWaddr 00:50:BA:8F:61:7A
          inet addr:192.168.0.2 Bcast:192.168.0.255 Mask:255.255.255.0
          inet6 addr: fe80::50:ba8f:617a/10 Scope:Link
            UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
            RX packets:1498792 errors:0 dropped:0 overruns:0 frame:0
            TX packets:1284980 errors:0 dropped:0 overruns:0 carrier:0
            collisions:1984 txqueuelen:100
            RX bytes:485691215 (463.1 Mb) TX bytes:123951388 (118.2 Mb)
            Interrupt:11 Base address:0xe800
```

Opcjonalnie: Konfigurowanie proxy

Jeśli korzystamy z proxy, musimy skonfigurować je w czasie instalacji. Jest to bardzo proste, wystarczy zdefiniować odpowiednią zmienną, zawierającą z informacją o serwerze proxy.

W większości przypadków można zdefiniować tę zmienną przy pomocy jego domeny. Pokażemy to na przykładzie serwera proxy.gentoo.org i portu 8080.

Listing 2: Definiowanie serwerów proxy

```
(Jeśli proxy filtruje HTTP)
# export http_proxy="http://proxy.gentoo.org:8080"
(Jeśli proxy filtruje FTP)
# export ftp_proxy="ftp://proxy.gentoo.org:8080"
(Jeśli proxy filtruje RSYNC)
# export RSYNC_PROXY="rsync://proxy.gentoo.org:8080"
```

Jeżeli proxy wymaga podania hasła i nazwy użytkownika, należy użyć następującej składni:

Listing 3: Dodawanie nazwy i hasła użytkownika do zmiennej

```
http://username:password@proxy.gentoo.org:8080
```

Testowanie sieci

Jeśli chcemy się upewnić, że pakiety dochodzą do celu, możemy spróbować pingowania któregoś z serwerów DNS (z pliku /etc/resolv.conf) lub dowolnie wybranej strony WWW.

Listing 4: Testowanie sieci

```
# ping -c 3 www.yahoo.com
```

Działa? Jeżeli tak, można pominąć resztę tego rozdziału i bezpośrednio przejść do rozdziału Przygotowanie dysków. Jeżeli nie, to pora zapoznać się z dalszą częścią tego tekstu.

3.b. Automatyczne konfigurowanie sieci

Niektóre media instalacyjne pozwalają na skorzystanie z narzędzia net-setup (dla typowych lub bezprzewodowych sieci) jeśli sieć nie zadziała od razu, pppoe-setup (dla użytkowników ASDL) albo pptp (dla użytkowników PPTP - dostępne tylko dla architektury x86, amd64, alpha, ppc oraz ppc64).

W przypadku gdy nośnik instalacyjny nie zawiera żadnego z wymienionych narzędzi, lub sieć wciąż nie funkcjonuje prawidłowo, należy przejść do akapitu Ręczna konfiguracja sieci.

- Użytkownicy typowych sieci Ethernet powinni przejść do podrozdziału Domyślnie: Używanie net-setup.
- Użytkownicy ASDL powinni przejść do paragrafu Alternatywnie: Używanie RP-PPPoE.
- A użytkownicy PPTP powinni przejść do części Alternatywnie: Używanie PPTP.

Domyślnie: Używanie net-setup

Najprostszą metodą konfigurowania sieci (poza automatyczną) jest ta zakładająca skorzystanie ze skryptu net-setup:

Listing 5: Uruchamianie skryptu net-setup

```
# net-setup eth0
```

Następnie należy udzielić odpowiedzi na serię dotyczących różnych parametrów sieci. Po zakończeniu wszystko powinno być skonfigurowane. Sprawdzamy połączenie tak jak opisano to wyżej. Jeśli wszystko działa to pora zacząć instalację Gentoo. Można pominąć resztę tego rozdziału i przejdź od razu do Przygotowywania dysków.

Jeśli sieć wciąż nie działa, przechodzimy do Ręcznej konfiguracji sieci.

Alternatywnie: Używanie RP-PPPoE

Jeśli do połączenia z Internetem potrzebne jest PPPoE, należy skorzystać z programu rp-ppoe nagranego na naszej płycie instalacyjnej. Skrypt pppoe-setup służy do konfiguracji połączenia. Zostaniemy zapytani o urządzenie sieciowe podłączone do modemu asdl, nazwę użytkownika i hasło, oraz o IP serwerów DNS i o to czy potrzebujemy podstawowego firewalla.

Listing 6: Używanie rp-pppoe

```
# pppoe-setup
# pppoe-start
```

Jeśli coś pojedzie nie tak, należy sprawdzić czy w /etc/ppp/pap-secrets lub /etc/ppp/chap-secrets podano prawidłową nazwę użytkownika i hasło oraz upewnić się, że wybrano właściwe urządzenie sieciowe. Jeśli nie zostało ono wykryte, konieczne będzie ręczne załadowanie odpowiednich sterowników. W takim wypadku należy przejść do Ręcznej konfiguracji sieci, gdzie szerzej to omówimy.

Jeżeli wszystko zadziałało przechodzimy do przygotowania dysków.

Alternatywnie: Używanie PPTP

Uwaga: Obsługa PPTP dostępna jest wyłącznie dla architektury x86.

Jeśli potrzebna jest obsługa PPTP, należy skorzystać z pptpclient zamieszczonego na płycie instalacyjnej. Najpierw jednak należy dodać prawidłową nazwę użytkownika i hasło do /etc/ppp/pap-secrets lub /etc/ppp/chap-secrets:

Listing 7: Edytowanie /etc/ppp/chap-secrets

```
# nano -w /etc/ppp/chap-secrets
```

Następnie konfigurujemy /etc/ppp/options.pptp:

Listing 8: Edytowanie /etc/ppp/options.pptp

```
# nano -w /etc/ppp/options.pptp
```

Po zakończeniu uruchamiamy program pptp (razem z niemożliwymi do ustawienia w options.pptp opcjami), aby połączyć się z serwerem:

Listing 9: Łączenie z serwerem dial-in

```
# pptp <server ip>
```

Kolejny etap instalacji to Przygotowywanie dysków.

3.c. Ręczne konfigurowanie sieci

Ładowanie odpowiednich modułów sieciowych

W czasie uruchamiania płyty instalacyjnej system spróbuje wykryć sprzęt i załadować odpowiednie sterowniki. W większości przypadków wykrywanie przebiega prawidłowo, czasem jednak trzeba ręcznie skorygować niektóre ustawienia.

Jeśli zawiódł net-setup lub pppoe-setup, możliwe, że nie została wykryta karta sieciowa. Oznacza to, że trzeba będzie ręcznie załadować odpowiedni sterownik.

Do wyświetlenia listy modułów kernela ze sterownikami dla urządzeń sieciowych używamy polecenia ls:

Listing 10: Szukanie modułów

```
# ls /lib/modules/`uname -r`/kernel/drivers/net
```

Gdy znajdziemy odpowiedni sterownik dla karty sieciowej, ładujemy go przy pomocy polecenia modprobe:

Listing 11: Ładowanie modułów kernela za pomocą modprobe

(W przykładzie załadujemy moduł pcnet32)

```
# modprobe pcnet32
```

Aby sprawdzić czy karta sieciowa została wykryta, korzystamy z polecenia ifconfig. Prawidłowy rezultat powinien wyglądać mniej więcej tak:

Listing 12: Sprawdzanie dostępności karty sieciowej. Wynik pozytywny

```
# ifconfig eth0
eth0      Link encap:Ethernet HWaddr FE:FD:00:00:00:00
          BROADCAST NOARP MULTICAST MTU:1500 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)   TX bytes:0 (0.0 b)
```

Następujący błąd oznacza, że karta nie została wykryta:

Listing 13: Sprawdzanie dostępności karty sieciowej. Wynik negatywny

```
# ifconfig eth0
eth0: error fetching interface information: Device not found
```

Jeżeli w komputerze znajduje się kilka kart sieciowych, będą one miały nazwy (kolejno) eth0, eth1, itp. Należy się upewnić, czy karta sieciowa której chcemy używać działa poprawnie i pamiętać o używaniu poprawnego nazewnictwa przy wykonywaniu czynności opisanych w dalszej części tego dokumentu. W Podręczniku zakładamy, że karta sieciowa nazywa się eth0.

Jeśli karta jest już prawidłowo rozpoznawana przez system, można ponownie użyć programów net-setup lub pppoe-setup (tym razem powinny zadziałać) lub skorzystać z poniższych instrukcji, aby połączenie skonfigurować ręcznie.

Następnie przechodzimy do jednej z następujących części:

- Używanie DHCP w celu uzyskania IP
- Przygotowywanie bezprzewodowego dostępu - dla kart bezprzewodowych
- Terminologia sieciowa, (niemal) wszystko co trzeba wiedzieć o działaniu sieci
- Używanie ifconfig i route - opis ręcznej konfiguracji sieci

Używanie DHCP

DHCP (Dynamic Host Configuration Protocol) umożliwia automatyczne otrzymywanie informacji o parametrach sieci (adresu IP, maski sieciowej, adresów broadcast, bramy, serwerów nazw, etc.). Niestety z metody tej można skorzystać tylko wtedy, gdy w sieci działa serwer DHCP (lub gdy ISP udostępnia taką usługę). Jeśli tak jest, można automatycznie skonfigurować połączenie przy pomocy dhcpcd.

Listing 14: Używanie dhcpcd

```
# dhcpcd eth0
Część administratorów wymaga używania
nazwy hosta i domeny dostarczanej przez serwer DHCP.
W tym przypadku używamy
# dhcpcd -HD eth0
```

Jeśli to zadziała (sprawdzamy pingując jakiś serwis internetowy, np. Google), wszystko jest gotowe i można pominąć resztę tego rozdziału i przejść bezpośrednio do Przygotowywania dysków.

Przygotowanie bezprzewodowego dostępu

Uwaga: Program iwconfig dostępny jest wyłącznie na płytach instalacyjnych dla architektur x86, amd64 oraz ppc. Opis instalacji dla pozostałych płyt znajduje się na stronach projektu linux-wlan-ng.

Jeśli używamy karty wireless (802.11), musimy ją skonfigurować. Aby poznać aktualne ustawienia skorzystamy z polecenia ifconfig. Rezultat wygląda zwykle tak:

Listing 15: Wyświetlanie aktualnych ustawień interfejsów kart do połączeń bezprzewodowych

```
# iwconfig eth0
eth0      IEEE 802.11-DS  ESSID:"GentooNode"
          Mode:Managed  Frequency:2.442GHz  Access Point: 00:09:5B:11:CC:F2
          Bit Rate:11Mb/s  Tx-Power=20 dBm  Sensitivity=0/65535
          Retry limit:16  RTS thr:off  Fragment thr:off
          Power Management:off
          Link Quality:25/10  Signal level:-51 dBm  Noise level:-102 dBm
          Rx invalid nwid:5901 Rx invalid crypt:0 Rx invalid frag:0 Tx
          excessive retries:237 Invalid misc:350282 Missed beacon:84
```

Uwaga: Część nazw urządzeń kart wireless to wlan0 zamiast eth0. Uruchomienie polecenia iwconfig bez dodatkowych parametrów pozwoli na poznanie nazwy odpowiedniego urządzenia.

W większości przypadków wystarcza zmodyfikowanie tylko dwóch opcji: ESSID (czyli nazwy sieci bezprzewodowej) oraz klucza WEP. Jeśli wyświetcone ESSID i Access Point są prawidłowe dla punktu dostępu i nie korzystamy z WEP, to połączenie już działa. Aby zmodyfikować ESSID lub dodać klucz WEP, skorzystamy z następujących poleceń:

Listing 16: Modyfikowanie ESSID i/lub dodawanie klucza WEP

```
(Polecenie zmieniające nazwę sieci na "GentooNode")
# iwconfig eth0 essid GentooNode
(Polecenie ustawiające heksadecymalny klucz WEPT)
# iwconfig eth0 key 1234123412341234abcd
(Polecenie ustawiające klucz ASCII - poprzedzony "s:")
# iwconfig eth0 key s:some-password
```

Można zatwierdzić te ustawienia ponownie wykonując iwconfig. Jeżeli sieć już działa, należy przejść do konfiguracji opcji na poziomie IP, opisanych w kolejnym paragrafie, (Terminologia sieciowa) lub wykorzystać omówiony wcześniej program net-setup.

Terminologia sieciowa

Uwaga: Znając adres IP, broadcast, maskę sieciową i serwery nazw, można pominąć tę część i od razu przejść do Używania ifconfig i route.

Jeżeli wszystkie powyższe zabiegi zawiodły, można jeszcze ręcznie skonfigurować sieć. Nie jest to bardzo trudne. Opiszemy najpierw różne parametry sieci, których znajomość jest konieczna. Opowiemy także o tym, czym jest brama, do czego służy maska sieciowa, jak ustala się adres broadcast i do czego potrzebne są serwery nazw.

Komputery w sieci są identyfikowane na podstawie adresów IP (Internet Protocol address). Każdy z nich jest kombinacją czterech liczb od 0 do 255. Cóż, przynajmniej my tak to widzimy. W rzeczywistości jest to ciąg 32 bitów (zer i jedynek). Pokażemy to na przykładzie:

Listing 17: Przykład adresu IP

Adres IP (liczby):	192.168.0.2
Adres IP (bity):	11000000 10101000 00000000 00000010

	192 168 0 2

Adres IP musi być unikalny dla każdego komputera, przynajmniej w obrębie jednej sieci. Aby oddzielić maszyny w sieci i poza nią IP podzielono na dwie części: część sieci oraz część hosta.

Podział zapisany jest za pomocą maski sieciowej, czyli zbioru zer poprzedzonego zbiorem jedynek. Ta część adresu, którą można odwzorować w jedynkach jest częścią sieci, reszta to część hosta. Zazwyczaj maskę zapisujemy jak zwykły adres IP.

Listing 18: Przykład oddzielenia sieci/hosta

Adres IP:	192 168 0 2
	11000000 10101000 00000000 00000010
Maska:	11111111 11111111 11111111 00000000
	255 255 255 0
+-----+-----+	
	Sieć Host

Innymi słowy, 192.168.0.14 wciąż jest częścią naszej przykładowej sieci, ale 192.168.1.2 już nie.

Adres broadcast składa się z części sieci takiej samej jak reszta komputerów oraz samych jedynek w części hosta. Każdy komputer nasłuchuje jego adresu IP, gdyż służy on do nadawania pakietów rozgłaszających.

Listing 19: Adres broadcast

Adres IP:	192 168 0 2
	11000000 10101000 00000000 00000010
Broadcast:	11000000 10101000 00000000 11111111
	192 168 0 255
+-----+-----+	
	Sieć Host

Żeby móc "surfować" po Internecie trzeba wiedzieć, który komputer udostępnia z nim połączenie. Komputer ten nazywamy bramą. To zwyczajna maszyna, ze zwykłym adresem IP (np. 172.168.0.1).

Poprzednio napisaliśmy, że każdy komputer ma własny adres IP. Aby móc się z nim połączyć za pomocą nazwy potrzebna jest usługa tłumacząca domeny (czyli na przykład dev.gentoo.org) na adresy IP (np. 64.5.62.82). Nazywa się ona serwerem nazw. Aby z niej skorzystać dodajemy ją do pliku /etc/resolv.conf.

Czasami brama może służyć również jako serwer nazw. Jeśli nie, to trzeba wpisać adresy DNS-ów dostarczanych przez ISP.

Podsumowując: potrzebne są następujące informacje:

Parametr	Przykład
Adres IP	192.168.0.2
Maska	255.255.255.0
Broadcast	192.168.0.255
Brama	192.168.0.1
Serwer(y) nazw	195.130.130.5, 195.130.130.133

Używanie ifconfig i route

Konfiguracja sieci składa się z trzech etapów. Najpierw przypisujemy sobie adres IP za pomocą ifconfig. Potem konfigurujemy bramę programem route. Na końcu wpisujemy adresy serwerów nazw do /etc/resolv.conf.

Aby przypisać komputerowi adres IP, należy oprócz niego znać również broadcast i maskę. Następnie wykonuje się następujące polecenie, zastępując wpisy \${IP_ADDR} swoim IP, \${BROADCAST} adresem broadcast i \${NETMASK} maską:

Listing 20: Używanie ifconfig

```
# ifconfig eth0 ${IP_ADDR} broadcast ${BROADCAST} netmask ${NETMASK} up
```

Następnie ustawiamy bramę polecienniem route. Wpis \${GATEWAY} należy zastąpić jej adresem IP:

Listing 21: Używanie route

```
# route add default gw ${GATEWAY}
```

Następnie otwieramy swoim ulubionym edytorem (w przykładzie skorzystamy z nano) plik /etc/resolv.conf:

Listing 22: Tworzenie /etc/resolv.conf

```
# nano -w /etc/resolv.conf
```

I wypełniamy go jak w przykładzie. Zamieniamy przy tym \${NAME SERVER1} oraz \${NAME SERVER2} adresami serwerów nazw:

Listing 23: Przykładowy /etc/resolv.conf

```
nameserver ${NAME SERVER1}
nameserver ${NAME SERVER2}
```

Na koniec testujemy sieć pingując jakiś serwer internetowy (na przykład Google). Jeśli wszystko działa, można rozpocząć instalację Gentoo, rozpoczynając od Przygotowywania dysków.

4. Przygotowywanie dysków

4.a. Wprowadzenie do urządzeń blokowych

Urządzenia blokowe

Rzućmy okiem na aspekty Gentoo Linux oraz ogólnie Linuksa związane z dyskami. Omówimy systemy plików, partycje oraz urządzenia blokowe. Następnie opiszemy proces podziału twardego dysku tak, aby jak najlepiej wykorzystać dostępne miejsce.

Zaczniemy od omówienia urządzeń blokowych. Najpopularniejszym z nich prawdopodobnie jest /dev/hda reprezentujący w Linuksie pierwszy napęd IDE. U posiadaczy urządzeń SCSI lub SATA pierwszym takim dyskiem jest /dev/sda.

Urządzenia blokowe stanowią abstrakcyjny interfejs dysków. Programy użytkownika mogą z nich korzystać nie martwiąc się o to czy napędy są typu IDE, SCSI czy jakiegoś innego. Przechowywane dane adresuje się jako ciąg 512-bajtowych bloków.

Partycje

Teoretycznie przeznaczenie na system całego dysku jest możliwe, zazwyczaj nie jest to jednak zbyt dobre rozwiązanie. Zamiast tego dzielimy napęd na mniejsze i dużo łatwiejsze w zarządzaniu urządzenia blokowe. W systemach opartych na architekturze x86 nazywane są one partycjami.

Wyróżniamy trzy rodzaje partycji: podstawowe, rozszerzone oraz logiczne.

Informacje o partycjach podstawowych przechowywane są w MBR (master boot record). Jako że jest on bardzo mały (512 bajtów), mieszczą się w nim dane maksymalnie czterech takich partycji (na przykład, od /dev/hda1 do /dev/hda4).

Specjalną odmianą partycji podstawowych są partycje rozszerzone (również obowiązuje je powyższy limit). Przechowują one wewnętrz siebie kolejne partycje. W ten sposób można ominąć niewygodną granicę i lepiej zagospodarować przestrzeń dyskową.

Partycje umieszczone i opisane wewnętrz rozszerzonych nazywamy logicznymi. Dane o nich nie znajdują się w MBR.

Zaawansowane metody przechowywania danych

Płyty instalacyjne dla architektury x86 posiadają obsługę EVMS oraz LVM2, co znacznie rozszerza możliwości partycjonowania dysków. W Podręczniku skupimy się na tworzeniu zwykłych partycji, warto jednak wiedzieć, że możliwe jest korzystanie również z nowocześniejszych rozwiązań.

4.b. Projektowanie schematu podziału

Domyślny schemat podziału

Można pominąć samodzielne rozrysowywanie schematu podziału dysku i po prostu skorzystać z naszego:

Partycja	System plików	Rozmiar	Opis
/dev/hda1	ext2	32M	Partycja rozruchowa
/dev/hda2	(swap)	512M	Partycja wymiany
/dev/hda3	ext3	Pozostałe wolne miejsce	Partycja główna

W dalszej części tekstu wyjaśnimy jak wiele i jak dużych partycji należy utworzyć. Można pominąć te informacje i przejść bezpośrednio do partycjonowania dysku przy pomocy fdisk.

Jak dużo i o jakim rozmiarze?

Ilość partycji ściśle zależy od danego środowiska. Na przykład, jeśli administrujesz systemem mającym wielu użytkowników, prawdopodobnie uznasz za stosowne oddzielenie /home aby poprawić bezpieczeństwo i uprościć tworzenie kopii zapasowych. Jeżeli docelowym zastosowaniem Gentoo jest serwer poczty, na osobnej partycji powinieneś umieścić /var, gdzie przechowywane są listy. Dobry wybór systemu plików może znacznie zwiększyć wydajność. Oddzielenie /opt jest dobrym rozwiązaniem na serwerach gier, gdyż większość używanego oprogramowania zostanie tam zainstalowana. Powód jest podobny jak przy /home: bezpieczeństwo i kopie zapasowe. Na pewno warto zapewnić dużo wolnego miejsca na /usr, ponieważ będą tam się znajdowały nie tylko dane wszystkich zainstalowanych pakietów, ale również ważąc 500 MB drzewo Portage i kody źródłowe programów.

Jak widać, wiele zależy od oczekiwany rezultatu. Rozdzielenie partycji ma wiele zalet:

- Daje możliwość wybrania dla poszczególnych partycji najbardziej wydajnego w danym zastosowaniu systemu plików.
- Zapełnienie całego wolnego miejsca na partycji przez wadliwie działający program nie będzie miało szkodliwego wpływu na całość systemu.
- Możliwe będzie skrócenie czasu kontroli systemów plików dzięki jednoczesnemu dokonywaniu jej na kilku partycjach (ma to znaczenie zwłaszcza na komputerze z wieloma twardymi dyskami).
- Montując część partycji lub woluminów z opcjami read-only (tylko do odczytu), nosuid (ignorowane są bity setuid), noexec (ignorowane są bity wykonywalności) itd. można znacznie poprawić bezpieczeństwo.

Niestety zbyt rozbudowany schemat podziału niesie również ze sobą spore problemy. Źle zaplanowany zaowocuje pustkami na zbyt dużych i ciasnotą na zbyt małych partycjach.

W przykładzie pokażemy partycjonowanie dysku o rozmiarze 20GB wykorzystywanego w laptopie z zainstalowanym serwerem poczty, stron internetowych oraz środowiskiem gnome:

Listing 1: Przykładowy podział dysku

```
$ df -h
Filesystem      Type  Size  Used  Avail Use% Mounted on
/dev/hda5        ext3  509M  132M  351M  28% /
/dev/hda2        ext3  5.0G  3.0G  1.8G  63% /home
/dev/hda7        ext3  7.9G  6.2G  1.3G  83% /usr
/dev/hda8        ext3  1011M 483M  477M  51% /opt
/dev/hda9        ext3  2.0G  607M  1.3G  32% /var
/dev/hda1        ext2   51M   17M   31M  36% /boot
/dev/hda6        swap   516M   12M   504M   2% <not mounted>
(Zostało 2GB nie przydzielonego do żadnej partycji miejsca - do wykorzystania w przyszłości)
```

/usr jest niemal w pełni zajęty (wykorzystane 83%), ale po instalacji wszystkich potrzebnych pakietów nie będzie się on zbytnio rozrastał. Przeznaczenie kilku gigabajtów na /var może wydać się zbyt rozrzutne. Należy jednak zwrócić uwagę, że Gentoo kompluje tam domyślnie wszystkie pakiety. Jeżeli chcemy, aby /var posiadało mniejszy rozmiar, na przykład 1GB, powinniśmy zmodyfikować zmienną PORTAGE_TMPDIR w pliku /etc/make.conf, tak aby prowadziła do partycji z odpowiednią ilością wolnego miejsca do komilacji ekstremalnie dużych pakietów takich jak OpenOffice.

4.c. Partycjonowanie dysku przy pomocy fdisk

Teraz przedstawimy proces tworzenia partycji wypisanych w naszym przykładowym schemacie podziału dysku. Oto on:

Partycja	Opis
/dev/hda1	Partycja rozruchowa
/dev/hda2	Partycja wymiany
/dev/hda3	Partycja główna

Ten schemat oczywiście można (i zwykle należy) zmodyfikować stosownie do własnych potrzeb.

Wyświetlanie aktualnego podziału dysku

fdisk to popularne i bardzo wygodne narzędzie służące do partycjonowania dysków. Zaczniemy od uruchomienia go dla naszego dysku (pokażemy to na przykładzie /dev/hda):

Listing 2: Uruchamianie fdisk

```
# fdisk /dev/hda
```

Po chwili ukaże się taki znak zachęty fdisk:

Listing 3: Znak zachęty fdisk

```
Command (m for help):
```

Aby wyświetlić obecną konfigurację partycji wpisujemy p:

Listing 4: Przykładowa konfiguracja partycji

Command (m for help): p

```
Disk /dev/hda: 240 heads, 63 sectors, 2184 cylinders
Units = cylinders of 15120 * 512 bytes
```

Device	Boot	Start	End	Blocks	Id	System
/dev/hda1		1	14	105808+	83	Linux
/dev/hda2		15	49	264600	82	Linux swap
/dev/hda3		50	70	158760	83	Linux
/dev/hda4		71	2184	15981840	5	Extended
/dev/hda5		71	209	1050808+	83	Linux
/dev/hda6		210	348	1050808+	83	Linux
/dev/hda7		349	626	2101648+	83	Linux
/dev/hda8		627	904	2101648+	83	Linux
/dev/hda9		905	2184	9676768+	83	Linux

Command (m for help):

Na prezentowanym dysku znajduje się siedem linuksowych systemów plików (każdy na osobnej partycji, oznaczonej jako "Linux") oraz partycja wymiany (oznaczona jako "Linux swap").

Usuwanie partycji

Zaczniemy od usunięcia starych partycji. Służy do tego polecenie d z odpowiednim dla każdej partycji numerem. Na przykład, aby pozbyć się /dev/hda1 należy wpisać:

Listing 5: Usuwanie partycji

```
Command (m for help): d
Partition number (1-4): 1
```

Partycja została zaznaczona do usunięcia. Nie będzie więcej pojawiać się po wpisaniu p, ale pozostałe nienaruszona dopóki zmiany nie zostaną fizycznie zapisane na dysku. Po popełnieniu błędu przy dzieleniu dysku wystarczy wpisać q i wcisnąć enter, aby wszystkie dokonane modyfikacje poszły w niepamięć.

Aby pozbyć się wszystkich istniejących partycji wpisujemy p, aby wyświetlić ich listę, a następnie po kolej kasujemy je poleceniem d z odpowiednim numerem. Na koniec p powinno dawać następujący rezultat:

Listing 6: Pusta tablica partycji

```
Disk /dev/hda: 30.0 GB, 30005821440 bytes
240 heads, 63 sectors/track, 3876 cylinders
Units = cylinders of 15120 * 512 = 7741440 bytes
```

Device	Boot	Start	End	Blocks	Id	System
--------	------	-------	-----	--------	----	--------

Command (m for help):

Kiedy już nasza tymczasowa kopia tablicy partycji będzie pusta możemy rozpocząć dzielenie dysku. Pokażemy to na przykładzie naszego domyślnego schematu, który należy odpowiednio zmienić przy każdej instalacji, tak aby możliwie najlepiej spełniał konkretne przed nim stawiane zadania.

Zakładanie partycji rozruchowej

Rozpoczniemy od utworzenia niewielkiej partycji rozruchowej. Wpisujemy n, aby ją założyć, a następnie p, aby nadać jej typ podstawowy i 1, ponieważ będzie to pierwsza taka partycja. Zapytani o pierwszy cylinder wciskamy enter, a przy pytaniu o ostatni wpisujemy +32M, by nadać jej rozmiar 32MB.

Listing 7: Zakładanie partycji rozruchowej

```
Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-3876, default 1): (Enter)
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-3876, default 3876): +32M
```

Teraz polecenie p powinno pokazywać taką listę:

Listing 8: Partycja rozruchowa

```
Command (m for help): p

Disk /dev/hda: 30.0 GB, 30005821440 bytes
240 heads, 63 sectors/track, 3876 cylinders
Units = cylinders of 15120 * 512 = 7741440 bytes

Device Boot      Start        End      Blocks   Id  System
/dev/hda1            1         14     105808+   83  Linux
```

Musimy również oznaczyć naszą partycję jako uruchamialną. W tym celu skorzystamy z polecenia a. Na kolejnych wydrukach podziału w kolumnie "Boot" przy naszej partycji pojawi się znak *.

Zakładanie partycji wymiany

Następnie tworzymy partycję wymiany. Ponownie korzystamy z polecenia n, potem wpisujemy p (gdyż ma to być partycja podstawowa) i 2, ponieważ będzie ona drugą tego typu. Zapytani o pierwszy cylinder wciskamy enter, a w odpowiedzi na pytanie o ostatni wpisujemy +512M (aby nadać partycji rozmiar 512MB). Na koniec wpisujemy t (aby zmienić jej typ), 2 (aby wybrać tę, którą właśnie stworzyliśmy) i 82 (co nada jej typ "Linux Swap"). Teraz polecenie p powinno pokazywać następującą listę:

Listing 9: Wydruk podziału po utworzeniu partycji wymiany

```
Command (m for help): p

Disk /dev/hda: 30.0 GB, 30005821440 bytes
240 heads, 63 sectors/track, 3876 cylinders
Units = cylinders of 15120 * 512 = 7741440 bytes

Device Boot      Start        End      Blocks   Id  System
/dev/hda1 *            1         14     105808+   83  Linux
/dev/hda2            15         81     506520    82  Linux swap
```

Zakładanie partycji głównej

Na koniec tworzymy partycję główną. Jeszcze raz posłużymy się poleceniem n. Zapytani o rodzaj wciskamy p (aby była partycją podstawową), następnie 3 (gdyż będzie już trzecią tego typu, czyli w naszym przypadku /dev/hda3), a na pytanie o pierwszy i ostatni cylinder wciskamy enter, dzięki czemu zajmie ona całą pozostałą wolną przestrzeń.

Ostatecznie polecenie p powinno pokazać następującą listę:

Listing 10: Wydruk podziału po utworzeniu partycji głównej

```
Command (m for help): p

Disk /dev/hda: 30.0 GB, 30005821440 bytes
240 heads, 63 sectors/track, 3876 cylinders
Units = cylinders of 15120 * 512 = 7741440 bytes

Device Boot Start End Blocks Id System
/dev/hda1 * 1 14 105808+ 83 Linux
/dev/hda2 15 81 506520 82 Linux swap
/dev/hda3 82 3876 28690200 83 Linux
```

Zapisywanie podziału partycji

Aby zachować ustalony podział i opuścić fdisk należy wpisać w.

Listing 11: Zachowywanie zmian i zamykanie fdisk

```
Command (m for help): w
```

Po utworzeniu partycji można przejść do zakładania systemów plików.

4.d. Zakładanie systemów plików

Wprowadzenie

Po utworzeniu partycji trzeba założyć na nich odpowiednie systemy plików. Osoby, którym wszystko jedno jaki system plików wybiorą lub te, które są zadowolone z domyślnych ustawień z Podręcznika mogą przejść do zakładania systemów plików na partycji. Pozostali muszą czytać dalej, aby dowiedzieć się więcej na ich temat.

Systemy plików?

Jądro Linuksa może współpracować z wieloma różnymi systemami plików. W Podręczniku omówimy te najczęściej używane, czyli ext2, ext3, ReiserFS i JFS.

ext2 to sprawdzony i popularny linuksowy system plików, którego główną wadą jest to, że nie posiada księgowania. Powoduje to, iż jego regularne kontrole przy starcie systemu bywają długotrwałe. Obecnie istnieją nowoczesne systemy plików z księgowaniem, które można szybko sprawdzić i to właśnie te polecamy naszym użytkownikom. Księgowanie zapobiega długotrwałym kontrolom podczas uruchamiania systemu oraz ewentualnym błędem spójności danych.

ext3 to odpowiednik ext2 posiadający księgowanie w trybach full oraz ordered, dzięki czemu w razie awarii dane odzyskiwane są błyskawicznie. Jest on bardzo dobrym i niezawodnym rozwiązaniem.

ReiserFS to system plików oparty na drzewie B*, oferujący dużą wydajność. Przy wielu małych plikach (poniżej 4k) może być szybszy od ext3 nawet piętnastokrotnie. ReiserFS jest wysoce skalowalny i posiada księgowanie, a począwszy od jądra 2.4.18, charakteryzuje go niezawodność i użyteczność zarówno na partycjach ogólnego przeznaczenia jak i w ekstremalnych przypadkach, takich jak ogromne partycje, operacje na wielu bardzo małych, lub bardzo dużych plikach czy też operacje na katalogach zawierających dziesiątki tysięcy plików.

XFS to system plików z księgowaniem, w pełni wspierany w Gentoo Linux przez jądro xfs-sources. Jest bardzo funkcjonalny i zoptymalizowany do skalowalności. Zalecamy go wyłącznie do systemów z nowoczesnymi dyskami SCSI i/lub ciągłego zapisu danych z nieprzerwanym dostępem zasilania. Ponieważ XFS przechowuje dużo danych w pamięci RAM, źle zaprojektowane programy (te nie zachowujące odpowiednich środków ostrożności podczas zapisywania plików na dysk, których niestety jest sporo) mogą doprowadzić w razie padu systemu do utraty danych.

JFS to bardzo wydajny system plików IBM'a wyposażony w księgowanie. Jest dość nowy i jest jeszcze za wcześnie by oceniać jego stabilność.

Zakładanie systemów plików na partycjach

Aby założyć na woluminie lub partycji system plików należy skorzystać z odpowiednich narzędzi:

System plików Program do jego zakładania

ext2	mke2fs
ext3	mke2fs -j
reiserfs	mkreiserfs
xfs	mkfs.xfs
jfs	mkfs.jfs

Na przykład, aby założyć ext2 na partycji boot (w naszym przypadku /dev/hda1) oraz ext3 na partycji root (w naszym przypadku /dev/hda3) powinieneś wykonać następujące polecenia:

Listing 12: Zakładanie systemu plików na partycji

```
# mke2fs /dev/hda1
# mke2fs -j /dev/hda3
```

Teraz stwórz systemy plików na swoich partycjach.

Aktywacja partycji wymiany

Aby utworzyć partycję wymiany, należy skorzystać z programu mkswap.

Listing 13: Tworzenie partycji wymiany

```
# mkswap /dev/hda2
```

Do aktywowania partycji wymiany używa się programu swapon:

Listing 14: Aktywacja partycji wymiany

```
# swapon /dev/hda2
```

Teraz tworzymy i aktywujemy partycję wymiany za pomocą wyżej podanych poleceń.

4.e. Montowanie

Po założeniu partycji i utworzeniu na nich systemów plików nadszedł czas na ich zamontowanie. Służy do tego program mount. Należy utworzyć odpowiednie katalogi dla montowanych partycji. W przykładzie zamontujemy partycję rozruchową i główną:

Listing 15: Montowanie partycji

```
# mount /dev/hda3 /mnt/gentoo
# mkdir /mnt/gentoo/boot
# mount /dev/hda1 /mnt/gentoo/boot
```

Uwaga: Aby przenieść /tmp na osobną partycję, należy po jego zamontowaniu odpowiednio zmienić prawa dostępu: chmod 1777 /mnt/gentoo/tmp. Dotyczy to również /var/tmp.

Konieczne będzie także zamontowanie systemu plików proc (wirtualny interfejs jądra) w katalogu /proc, ale najpierw musimy umieścić odpowiednie pliki na partycjach.

5. Wypakowywanie plików instalacyjnych Gentoo

5.a. Instalowanie tarballa stage

Ustawienie poprawnej daty i czasu

Na samym początku całego procesu instalacji należy sprawdzić datę/czas i ewentualnie je zaktualizować. Niezsynchonizowany zegar może być przyczyną dziwnych błędów w przeszłości!

Aby zweryfikować aktualną datę/czas, uruchamiamy date:

Listing 1: Sprawdzenie daty/czasu

```
# date
nie sie 21 01:56:26 UTC 2005
```

Jeżeli wyświetlane data i czas są złe, musimy je uaktualnić polecienniem date MMDDggmmRRRR (Miesiąc, Dzień, godzina, minuta, i Rok). Na tym etapie powinniśmy korzystać z czasu UTC. W późniejszym czasie będziemy mogli zdefiniować naszą strefę czasową. Na przykład, aby ustawić datę 29 marca 2005 roku, 16:21:

Listing 2: Ustawienie daty/czasu UTC

```
# date 032916212005
```

Podejmowanie decyzji

W następnym kroku należy wykonać instalację wybranego tarballa stage3. Można go pobrać z Internetu lub, jeśli działamy z którejś płytka Gentoo Universal Installation CD, przekopiować z CD. Jeżeli mamy Universal CD i na płycie znajduje się stage którego chcemy używać ściąganie go z Internetu jest tylko niepotrzebną stratą czasu, gdyż pliki stage są takie same.

- Domyślnie: Użycie stage z Internetu.
- Alternatywnie: Wykorzystanie stage z płyty instalacyjnej.

5.b. Domyślnie: Użycie stage z Internetu

Pobieranie tarballa stage

Na początku przechodzimy do punktu montowania systemu plików Gentoo (zwykle jest to /mnt/gentoo):

Listing 3: Przechodzenie do punktu montowania systemu plików Gentoo

```
# cd /mnt/gentoo
```

W zależności od medium instalacyjnego mamy do dyspozycji kilka narzędzi, za pomocą których możemy pobrać plik stage. Jeżeli mamy program links możemy wejść bezpośrednio na [listę serwerów lustrzanych Gentoo](#) i wybrać serwer, który znajduje się najbliżej.

Jeżeli nie mamy programu links, musimy skorzystać z przeglądarki lynx do tego celu. Aby używać serwera proxy musimy również wyeksportować zmienne http_proxy i ftp_proxy:

Listing 4: Ustawienie informacji o proxy dla lynxa

```
# export http_proxy="http://proxy.server.com:port"
# export ftp_proxy="http://proxy.server.com:port"
```

W dalszej części zakładamy, że do swojej dyspozycji mamy przeglądarkę links.

Przechodzimy do releases/, a następnie do katalogu odpowiedniej architektury (np. x86/) oraz wersji Gentoo (2006.1/ lub) i na koniec do stages/. Znajdują się tam pliki stage dla wybranej architektury (czasem mogą być umieszczone w podkatalogach dla różnych podarchitektur). Wybieramy jeden i naciskamy D, aby go ściągnąć. Kiedy pobieranie dobiegnie końca wciskamy Q, aby wyjść z przeglądarki.

Listing 5: Przeglądanie listy serwerów lustrzanych za pomocą links

```
# links http://www.gentoo.org/main/en/mirrors.xml
(Jeżeli potrzebne jest proxy w links)
# links -http-proxy serwer.proxy.com:8080 http://www.gentoo.org/main/en/mirrors.xml
```

Wybranie pliku stage3 jest konieczne, nie wspieramy już instalacji za pomocą stage 1 i 2.

Jeśli chcemy zweryfikować poprawność pobranych archiwów stage, musimy porównać wynik polecenia md5sum z sumami MD5 udostępnianymi na serwerze. Np. aby sprawdzić poprawność tarballa stage dla architektury x86:

Listing 6: Sprawdzanie integralności archiwum stage

```
# md5sum -c stage3-x86-2006.1.tar.bz2.DIGESTS
stage3-x86-2006.1.tar.bz2: OK
```

Rozpakowywanie tarballa Stage

Wypakujemy pobrany plik stage przy pomocy programu tar:

Listing 7: Wypakowanie stage

```
# tar xvjpf stage3-*.tar.bz2
```

Należy użyć dokładnie tych samych przełączników (xvjpf). Opcja x oznacza wypakuj, v to wyświetl, aby widzieć co się dzieje podczas wypakowywania (ok, to jest opcjonalne), j służy do dekompresji archiwum bzip2, p to zachowuj uprawnienia, natomiast f podkreśla, że chcemy rozpakować to, co czytamy z pliku, a nie ze standardowego wejścia.

Uwaga: Obrazy płyt instalacyjnych niektórych architektur (np. MIPS) zawierają tar wbudowany w BusyBox, który aktualnie nie posiada opcji v. W takim wypadku należy skorzystać z parametrów xjpf.

Gdy stage jest już zainstalowany, pora przejść do Instalacji Portage.

5.c. Alternatywnie: Wykorzystanie stage z płyty instalacyjnej

Rozpakowanie tarballa stage

Ważne: Jeśli korzysta się z płyty dla architektury x86 i instalatora Gentoo, należy pobrać archiwa z Internetu, ponieważ nie znajdują się na płytach instalacyjnych.

Pliki stage umieszczone są na CD w katalogu /mnt/cdrom/stages. Aby obejrzeć ich spis korzystamy z polecenia ls:

Listing 8: Lista dostępnych wersji stage

```
# ls /mnt/cdrom/stages
```

Jeśli system zgłasza błąd to możliwe, że musimy najpierw zamontować CD-ROM:

Listing 9: Montowanie CD-ROM

```
# ls /mnt/cdrom/stages
ls: /mnt/cdrom/stages: No such file or directory
# mount /dev/cdrom0 /mnt/cdrom
# ls /mnt/cdrom/stages
```

Teraz przechodzimy do punktu montowania Gentoo (zwykle /mnt/gentoo):

Listing 10: Zmiana katalogu na /mnt/gentoo

```
# cd /mnt/gentoo
```

Następnie wypakowujemy wybrany tarball. Użyjemy do tego celu programu tar. Przelączniki (-xvjpf) muszą być takie same! Należy pamiętać, że argument v jest opcjonalny i nie jest obsługiwany przez pewne wersje programu tar. W kolejnym przykładzie wykorzystujemy plik stage3-<podarchitektura>-2006.1.tar.bz2. Oczywiście jego nazwę należy odpowiednio zmodyfikować.

Listing 11: Wypakowanie tarballa stage

```
# tar xvjpf /mnt/cdrom/stages/stage3-<architektura>-2006.1.tar.bz2
```

Gdy stage zostanie zainstalowany, przechodzimy do Instalacji Portage.

5.d. Instalacja Portage

Wypakowanie snapshota Portage

W tym rozdziale omówimy proces instalacji snapshota Portage, kolekcji plików, które informują Portage jakie programy można zainstalować, które profile są dostępne itp.

Ściąganie i instalowanie snapshota Portage

Przechodzimy do miejsca gdzie zamontowaliśmy system plików (zwykle /mnt/gentoo):

Listing 12: Przechodzenie do punktu montowania Gentoo

```
# cd /mnt/gentoo
```

Uruchamiamy links (lub lynx) i przechodzimy do listy mirrorów Gentoo. Wybieramy jeden z serwerów, najlepiej jak najbliższy naszej lokalizacji i przechodzimy do katalogu snapshots/. Ściągamy najnowszy snapshot Portage (portage-latest.tar.bz2) poprzez jego wybór i naciśnięcie klawisza D.

Listing 13: Przeglądanie listy mirrorów Gentoo

```
# links http://www.gentoo.org/main/en/mirrors.xml
```

Teraz wychodzimy z przeglądarki naciskając klawisz Q. Plik znajduje się w katalogu /mnt/gentoo.

Jeśli chcemy sprawdzić poprawność ściagniętego pliku, należy użyć md5sum i porównać sumę kontrolną pliku z podaną na serwerze lustrzanym.

Listing 14: Sprawdzanie poprawności pliku

```
# md5sum -c portage-latest.tar.bz2.md5sum
portage-latest.tar.bz2: OK
```

W następnym kroku wypakujemy snapshot Portage. Należy użyć dokładnie tych samych poleceń; ostatnia opcja to duża litera C, nie małe c.

Listing 15: Wypakowywanie snapshota Portage

```
# tar xvjf /mnt/gentoo/portage-latest.tar.bz2 -C /mnt/gentoo/usr
```

5.e. Konfigurowanie opcji komplikacji

Wprowadzenie

Jest wiele możliwych do skonfigurowania zmiennych wpływających na zachowanie Gentoo. Możemy je wprowadzać jako zmienne środowiskowe (poprzez export), ale wtedy nie zostaną zapisane na stałe. Zamiast tego Portage do zachowywania konfiguracji używa pliku konfiguracyjnego /etc/make.conf. Pora wziąć się za jego edycję.

Uwaga: Opatrzona komentarzami lista wszystkich możliwych zmiennych znajduje się w pliku /mnt/gentoo/etc/make.conf.example. Do szczęśliwego ukończenia instalacji wystarczy wyedytowanie tylko kilku z nich, tych, których listę przedstawiamy poniżej.

Uruchamiamy ulubiony edytor (w przykładach używamy nano), którym wprowadzimy omawiane nieco dalej opcje optymalizacji.

Listing 16: Edytowanie /etc/make.conf

```
# nano -w /mnt/gentoo/etc/make.conf
```

Plik make.conf.example ma charakterystyczną strukturę: linie z komentarzem rozpoczynają się od znaku "#", linie zawierające zmienne używają składni ZMIENNA="zawartość". Takiej samej składni używa także plik /etc/make.conf. Kilka z tych zmiennych zostało przedyskutowanych poniżej.

CHOST

Zmienna CHOST określa architekturę, na jakiej będzie budowany system i powinna już być ustawiona na odpowiednią wartość. **Nie wolno jej zmieniać**, ponieważ może to zepsuć system. Jeśli w zmiennej CHOST znajduje się nieprawidłowa wartość, najprawdopodobniej oznacza to, że użyto złego archiwum stage3.

CFLAGS i CXXFLAGS

Zmienne CFLAGS i CXXFLAGS definiują flagi optymalizujące używane odpowiednio przez kompilator gcc C i C++. Choć generalnie określamy ich wartości tutaj, maksimum wydajności osiągniemy dopasowując je do każdego programu z osobna. Jest tak dlatego, że programy znaczco różnią się między sobą.

W make.conf należy zdefiniować flagi optymalizacji co do których jesteśmy przekonani, że w głównej mierze poprawią czas reakcji systemu. Nie przypisujemy pod tą zmienną ustawień eksperymentalnych; przesada w optymalizacji może spowodować, że programy zaczną źle funkcjonować (nagle przerywać działanie, lub nawet gorzej, wcale nie działać).

Nie będziemy tłumaczyć znaczenia wszystkich możliwych opcji optymalizacji. Wszystkie są wymienione w Podręczniku Online GNU lub stronę info gcc (info gcc -- działa tylko na systemach linuksowych). Plik make.conf.example sam zawiera dużo informacji i przykładów - należy go uważnie przeczytać.

Pierwszym ustawieniem jakim się tu zajmiemy jest flaga -march=, która określa docelową architekturę. Możliwe jej wartości są opisane jako komentarze w make.conf.example. Na przykład dla architektury x86 Athlon XP będzie to:

Listing 17: Ustawienie GCC march

```
# Użytkownicy AMD64, którzy chcą mieć natywny system 64bit powinni używać flagi -march=k8
# Użytkownicy EM64T powinni wybrać march=nocona
-march=athlon-xp
```

Drugim jest flaga -O (to jest duże O, nie zero), która określa klasę optymalizacji gcc. Dostępne klasy to s (optymalizacja rozmiaru), 0 (brak optymalizacji), 1, 2 lub 3 - coraz silniej optymalizujące (każda z nich używa tych samych flag, co poprzednia oraz dodaje własne). Jako przykład posłuży nam klasa optymalizacji 2:

Listing 18: Ustawienia optymalizacji poprzez GCC

```
-O2
```

Inne popularne flagi optymalizujące to -pipe (gcc używa potoków zamiast plików tymczasowych w komunikacji między różnymi etapami komplikacji) oraz -fomit-frame-pointer (w rejestrach nie będą przechowywane wskaźniki ramki dla funkcji, które ich nie wymagają).

Używanie flagi -fomit-frame-pointer może powodować poważne problemy podczas debugowania kodu!

Podczas definiowania CFLAGS i CXXFLAGS można łączyć kilka flag optymalizacji, na przykład w ten sposób:

Listing 19: Definiowanie zmiennych CFLAGS i CXXFLAGS

```
CFLAGS="-march=athlon-xp -pipe -O2"      # Użytkownicy AMD64 powinni używać flagi march=k8  
CXXFLAGS="${CFLAGS}"                      # Użytkownicy EM64T powinni wybrać march=nocona  
                                         # Użycie tych samych ustawień dla obu zmiennych
```

MAKEOPTS

Za pomocą MAKEOPTS definiujemy jak wiele równoległych komplikacji będzie przeprowadzanych podczas przygotowywania pakietu do instalacji. Sugerowaną liczbą jest ilość procesorów w systemie powiększona o jeden, nie jest to jednak zawsze najlepsze wyjście.

Listing 20: MAKEOPTS dla przeciętnego systemu jednoprocesorowego

```
MAKEOPTS="-j 2"
```

Gotowi, do biegu, start!

Na koniec poprawiamy jeszcze odrobinę /mnt/gentoo/etc/make.conf i zapisujemy wyniki naszych prac (w nano za pomocą Ctrl-X). Teraz jesteśmy przygotowani na Instalację systemu podstawowego .

6. Instalowanie systemu podstawowego

6.a. Praca w chroot

Opcjonalnie: Wybieranie serwerów lustrzanych

Aby móc szybko ściągać źródła programów, należy wybrać szybki serwer lustrzany. Portage używa serwerów zawartych w zmiennej GENTOO_MIRRORS, która znajduje się w pliku make.conf. Aby wybrać najlepsze serwery, należy wejść na stronę listy serwerów lustrzanych Gentoo i wybrać z nich te, które znajdują się najbliżej lub użyć narzędzia mirrorselect, które potrafi w prosty sposób automatycznie wybrać najlepsze serwery lustrzane.

Listing 1: Sposób użycia programu mirrorselect

```
# mirrorselect -i -o >> /mnt/gentoo/etc/make.conf
```

Ostrzeżenie: Nie należy wybierać żadnych serwerów IPv6. Nasze pliki stage aktualnie nie wspierają IPv6.

Następną ważną sprawą jest ustawienie zmiennej SYNC w pliku make.conf. Zmienna ta wskazuje na serwer rsync, z którego będzie aktualniane drzewo Portage (kolekcja ebuildów, czyli skryptów które zawierają wszystkie informacje potrzebne do ściągnięcia i zainstalowania programów). Można ręcznie wpisać serwer, którego chcemy używać lub skorzystać z programu mirrorselect:

Listing 2: Wybór serwera rsync za pomocą programu mirrorselect

```
# mirrorselect -i -r -o >> /mnt/gentoo/etc/make.conf
```

Po konfiguracji plików za pomocą programu mirrorselect należy sprawdzić czy wszystko zostało prawidłowo dopisane do pliku /mnt/gentoo/etc/make.conf.

Kopiowanie informacji o DNS

Zanim zmienimy środowisko pracy, musimy wykonać pewną bardzo ważną czynność. Jest nią przekopiowanie ustawień DNS z pliku /etc/resolv.conf do nowego środowiska. Jest to konieczne, by sieć działała także tam. Plik /etc/resolv.conf określa jakie serwery nazw będą używane dla sieci.

Listing 3: Kopiowanie informacji o DNS

(Opcja "-L" jest konieczna, sprawia, że nie zostanie skopiowane dowiązanie symboliczne)
`# cp -L /etc/resolv.conf /mnt/gentoo/etc/resolv.conf`

Montowanie systemów plików /proc i /dev

Następnie przemontowujemy system plików /proc do /mnt/gentoo/proc, aby umożliwić systemowi korzystanie z informacji dostarczanych przez jądro także w środowisku chrootowanym oraz ponownie montujemy system plików /dev.

Listing 4: Montowanie /proc i /dev

```
# mount -t proc none /mnt/gentoo/proc
# mount -o bind /dev /mnt/gentoo/dev
```

Zmiana środowiska

Teraz, gdy wszystkie partie są już założone, a podstawowe środowisko zainstalowane, nadszedł czas wejścia do niego poprzez chroot. Oznacza to przejście z systemu instalacyjnego (płyty instalacyjnej lub innego medium) do systemu instalowanego (czyli na założone partie).

Przechodzenie odbywa się w trzech etapach. Najpierw zmieniamy katalog z / (w systemie instalacyjnym) na /mnt/gentoo (na założonych partiach) poleceniem chroot. Następnie tworzymy nowe środowisko przy pomocy polecenia env-update, które wyeksportuje nowe zmienne środowiskowe. Ostatecznie wczytujemy te zmienne do pamięci poleceniem source.

Listing 5: Zmiana środowiska poprzez chroot

```
# chroot /mnt/gentoo /bin/bash
# env-update
>> Regenerating /etc/ld.so.cache...
# source /etc/profile
# export PS1="(chroot) $PS1"
```

Gratulacje! Znajdujemy się wewnątrz nowego systemu Gentoo Linux. Oczywiście do końca jeszcze daleko, przecież zostało jeszcze kilka rozdziałów Podręcznika do przeczytania. :-)

6.b. Konfiguracja Portage

Aktualizacja drzewa Portage

Aktualizujemy drzewo Portage za pomocą polecenia emerge --sync.

Listing 6: Aktualizowanie drzewa Portage

```
# emerge --sync
(Jeśli korzysta się z wolnych terminali, takich jak konsola bufora
ramki, należy dodać parametr --quiet dla przyspieszenia całego procesu)
# emerge --sync --quiet
```

Portage używa protokołu RSYNC do uaktualniania drzewa pakietów. Jeżeli powyższe polecenie zakończy się niepowodzeniem (np. z winy firewalla) używamy polecenia emerge-webrsync, które ściąga i instaluje drzewo Portage przy za pomocą protokołu HTTP.

Jeśli otrzymamy ostrzeżenie, że dostępna jest nowa wersja programu Portage i należy dokonać jego aktualizacji, należy to natychmiast zrobić. Dokonuje się tego poleceniem emerge portage.

Wybór odpowiedniego profilu

Najpierw mała definicja.

Profil jest szablonem budowy systemu Gentoo. Nie tylko określa domyślne wartości dla CHOST, CFLAGS i innych ważnych zmiennych, ale również ogranicza wersje pakietów jakie mogą zostać zainstalowane w systemie. Wszystkie te informacje są konfigurowane przez deweloperów Gentoo.

Dawniej użytkownik w bardzo małym stopniu miał wpływ na profil. Jednakże ostatnio użytkownicy architektur x86, hppa i alpha mogą wybierać między dwoma profilami, jednym dla jąder serii 2.4 i drugim dla jąder serii 2.6. Wymaganie takie wynikło z potrzeby lepszej integracji jąder serii 2.6. Architektury ppc i ppc64 mają kilka różnych profili do wyboru, szczegółowo omówimy je później.

Aktualnie używany profil można sprawdzić za pomocą poniższego polecenia:

Listing 7: Sprawdzanie profilu systemowego

```
# ls -FGg /etc/make.profile
lrwxrwxrwx 1 48 Apr 8 18:51 /etc/make.profile -> ../../usr/portage/profiles/default-
linux/x86/2006.1/
```

Jeśli korzysta się z jednej z trzech wymienionych wyżej architektur, domyślnym profilem będzie ten korzystający z jądra 2.6. Jest to rekommendowana wartość domyślna, ale użytkownicy mają również możliwość wybrania profilu z jądrem 2.4.

Instalacja systemu opartego o jądro 2.4 jest możliwa tylko wtedy, gdy istnieje odpowiedni do tego profil. Można sprawdzić to w następujący sposób (przykład dla architektury x86):

Listing 8: Sprawdzanie, czy istnieje dodatkowy profil

```
# ls -d /usr/portage/profiles/default-linux/x86/no-nptl/2.4
/usr/portage/profiles/default-linux/x86/no-nptl/2.4
```

Powyższy przykład wskazuje na to, że istnieje dodatkowy profil dla jądra 2.4 (nie pojawił się błąd związany z nieistniejącym katalogiem).

Zalecamy pozostanie przy domyślnym profilu, jeśli jednak konieczne jest skorzystanie z 2.4, można go ustawić w następujący sposób:

Listing 9: Przechodzenie na profil 2.4

```
(Jeśli to konieczne, x86 ponizej należy zastąpić nazwą używanej architektury)
# ln -snf /usr/portage/profiles/default-linux/x86/no-nptl/2.4 /etc/make.profile
(Lista plików profilu 2.4)
# ls -FGg /etc/make.profile/
total 12
-rw-r--r-- 1 939 Dec 10 14:06 packages
-rw-r--r-- 1 347 Dec 3 2004 parent
-rw-r--r-- 1 573 Dec 3 2004 virtuals
```

W wydaniu 2006.1 pojawiła się cała seria nowych profili dla architektury ppc.

Listing 10: Profile dla PPC

```
(Wspólny dla wszystkich podarchitektur
główny profil PPC, wersja minimalna)
# ln -snf /usr/portage/profiles/default-linux/ppc/ppc32/2006.1 /etc/make.profile
(Profil G3)
# ln -snf /usr/portage/profiles/default-linux/ppc/ppc32/2006.1/G3 /etc/make.profile
(Profil G3 Pegasos)
# ln -snf /usr/portage/profiles/default-linux/ppc/ppc32/2006.1/G3/Pegasos/
/etc/make.profile
(Profil G4 (Altivec))
# ln -snf /usr/portage/profiles/default-linux/ppc/ppc32/2006.1/G4 /etc/make.profile
(Profil G4 (Altivec) Pegasos)
# ln -snf /usr/portage/profiles/default-linux/ppc/ppc32/2006.1/G4/Pegasos/
/etc/make.profile
```

Architektura ppc64 od wydania 2006.1 również posiada wiele profili:

Listing 11: Profile dla PPC64

```
(Wspólny 64-bitowy profil dla wszystkich komputerów PPC64)
# ln -snf /usr/portage/profiles/default-linux/ppc/ppc64/2006.1/64bit-userland /etc/make.profile
(Wspólny 32-bitowy profil dla wszystkich komputerów PPC64)
# ln -snf /usr/portage/profiles/default-linux/ppc/ppc64/2006.1/32bit-userland /etc/make.profile
(Każdy z profili posiada podprofile, należy zastąpić wpis (userland) jedną z powyższych pozycji)
(Profil 970 dla JS20)
# ln -snf /usr/portage/profiles/default-linux/ppc/ppc64/2006.1/(userland)/970 /etc/make.profile
(Profil G5)
# ln -snf /usr/portage/profiles/default-linux/ppc/ppc64/2006.1/(userland)/970/pmac
/etc/make.profile
(Profil POWER3)
# ln -snf /usr/portage/profiles/default-linux/ppc/ppc64/2006.1/(userland)/power3 /etc/make.profile
(Profil POWER4)
# ln -snf /usr/portage/profiles/default-linux/ppc/ppc64/2006.1/(userland)/power4 /etc/make.profile
(Profil POWER5)
# ln -snf /usr/portage/profiles/default-linux/ppc/ppc64/2006.1/(userland)/power5 /etc/make.profile
(Profil multilib nie jest jeszcze stabilny w tym wydaniu)
```

Konfiguracja zmiennych USE

USE to jedna z najważniejszych zmiennych w Gentoo. Niektóre programy mogą być komplikowane z dodatkową obsługą niektórych funkcji lub bez niej. Na przykład możliwe jest budowanie różnych programów ze wsparciem dla bibliotek gtk lub qt. Inne pakiety możemy z kolei wyposażyć w obsługę SSL, bądź też jej pozbawić. Jeszcze inne mogą być komplikowane ze wsparciem bufora ramki (svgalib) zamiast X11 (serwera X).

Większość dystrybucji kompiluje swoje pakiety ze wsparciem dla tak wielu elementów, jak to tylko możliwe, powiększając rozmiar programów i czas ich uruchamiania, nie wspominając o olbrzymiej liczbie zależności.

W Gentoo możemy zdecydować, z którymi opcjami dany pakiet powinien być budowany. I to właśnie jest moment, kiedy USE wkracza do gry.

W zmiennych USE definiujemy słowa kluczowe zamieniane następnie na opcje komplikowania. Na przykład dodanie do zmiennej ssl włączy obsługę SSL w programach, które go wykorzystują. -X usunie wsparcie dla serwera X (należy zwrócić uwagę na znak minusa z przodu). Ustawienie gnome gtk -kde -qt zaowocuje wsparciem dla GNOME (oraz gtk), ale nie dla KDE (i związanym z nim ścisłe qt), znakomicie przygotowując grunt pod GNOME.

Domyślny zestaw flag USE znajduje się w pliku make.defaults wybranego profilu. Wszystkie pliki make.defaults znajdują się w katalogu wskazywanym przez dowiązanie /etc/make.profile oraz w katalogach nadzędnych. Aktualna konfiguracja USE jest zawsze sumą wszystkich flag ustawionych w plikach make.defaults. Wszystko co umieścimy w pliku /etc/make.conf zostanie dodane do tej zmiennej. Jeśli chcemy coś z niej usunąć wpisujemy wybraną flagę ze znakiem minus na początku. Nie wolno zmieniać plików wewnętrz katalogu /etc/make.profile, zmiany zostaną nadpisane przy następnej aktualizacji drzewa Portage.

Pełny opis USE znajduje się w drugiej części Podręcznika Gentoo, w rozdziale Flagi USE. Kompletną charakterystykę dostępnych flag USE znajdziemy w pliku /usr/portage/profiles/use.desc.

Listing 12: Przegląd dostępnych flag USE

```
# less /usr/portage/profiles/use.desc
(Używamy strzałek, aby przewijać plik. Aby wyjść naciskamy 'q')
```

Jako przykład przedstawimy flagi USE dla systemu bazującego na KDE ze wsparciem dla DVD, ALSA i nagrywania CD:

Listing 13: Edytowanie /etc/make.conf

```
# nano -w /etc/make.conf
```

Listing 14: Ustawienia USE

```
USE="-gtk -gnome qt kde dvd alsa cdr"
```

Opcjonalnie: Lokalizacje GLIBC

Zwykle w systemie używa się tylko jednej, góra dwóch lokalizacji. Można je wybrać w pliku /etc/locale.gen.

Listing 15: Otwieranie pliku /etc/locale.gen

```
# nano -w /etc/locale.gen
```

Poniższy przykład to lokalizacje polskie oraz angielskie (Ameryka) z obsługą kodowania znaków (jak UTF-8).

Listing 16: Ustawianie lokalizacji

```
en_US ISO-8859-1
en_US.UTF-8 UTF-8
pl_PL ISO-8859-2
pl_PL.UTF-8 UTF-8
```

Następnie należy uruchomić polecenie locale-gen, które utworzy wszystkie lokalizacje wybrane w pliku /etc/locale.gen.

Uwaga: locale-gen jest dostępne w glibc-2.3.6-r4 i nowszych. Jeśli jest zainstalowana starsza wersja glibc, należy ją teraz zaktualizować.

Kolejny etap instalacji to Konfigurowanie jądra.

7. Konfigurowanie jądra

7.a. Strefa czasowa

System musi wiedzieć gdzie się znajduje, musimy zatem poprawnie skonfigurować jego ustawienia strefy czasowej. Zaczynamy od wyszukania odpowiedniej dla nas strefy w katalogu /usr/share/zoneinfo, a następnie skopiujemy ją do /etc/localtime. Należy unikać stref czasowych o nazwie /usr/share/zoneinfo/Etc/GMT*, ponieważ ich nazwy mogą być mylące, na przykład GMT-8 jest w rzeczywistości GMT+8.

Listing 1: Ustawianie strefy czasowej

```
# ls /usr/share/zoneinfo
(przy założeniu, że nasza strefa to "Poland")
# cp /usr/share/zoneinfo/Poland /etc/localtime
```

7.b. Instalowanie źródeł

Wybór jądra

Jądro Linux jest sercem każdej dystrybucji i stanowi interfejs pomiędzy programami użytkownika, a sprzętem. Gentoo pozwala użytkownikom na wybranie spośród kilku różnych jego źródeł. Pełna ich lista wraz z opisami znajduje się w dokumencie Gentoo Linux Kernel Guide.

Dla architektury x86 dostępne są następujące jądra: vanilla-sources (standardowe źródła serii 2.4 rozwijane przez deweloperów Linuksa), oraz nasze gentoo-sources (źródła z poprawkami zwiększącymi wydajność).

Wybrane źródła instaluje się przy pomocy polecenia emerge. Flaga USE="-doc" uniemożliwi zainstalowanie xorg-x11 i innych zbędnych na tym etapie zależności. Flaga USE="symlink" nie jest konieczna w przypadku nowej instalacji, jednak zapewnia w późniejszym okresie tworzenie prawidłowych dowiązań symbolicznych /usr/src/linux.

Listing 2: Instalowanie źródeł jądra

```
# USE="-doc symlink" emerge gentoo-sources
```

W katalogu /usr/src, pod nazwą linux, powinno znajdować się mniej więcej takie dowiązanie symboliczne wskazujące na aktualne źródła. W tym wypadku wskazuje na źródła gentoo-sources-2.6.12-r10. W komputerze użytkownika może być to inna wersja, dlatego należy mieć to na uwadze.

Listing 3: Podgląd dowiązania symbolicznego do źródeł jądra

```
# ls -l /usr/src/linux
lrwxrwxrwx    1 root      root     12 Oct 13 11:04 /usr/src/linux -> linux-2.6.17-r5
```

Pora na skonfigurowanie i skompilowanie źródeł jądra. Można użyć do tego programu genkernel, który zbuduje uniwersalne jądro, takie jak to znajdujące się na płytach instalacyjnych. Można również przeprowadzić cały proces ręcznie, co pozwoli na lepsze dostosowanie jądra do indywidualnych potrzeb użytkownika. Najpierw omówimy tę drugą, znacznie lepszą metodę.

Proces ręcznej konfiguracji jądra został opisany w paragrafie Domyślnie: ręczna konfiguracja. Opis użycia genkernela znajduje się w części Alternatywnie: użycie genkernel.

7.c. Domyślnie: Ręczna konfiguracja

Wprowadzenie

Ręczna konfiguracja jądra to zwykle najtrudniejsze zadanie przed jakim musi stać każdy początkujący użytkownik Linuksa. Zwykle po skompilowaniu kilku zapomina się, że kiedykolwiek uważało się to za trudną czynność. :-)

Nie sposób jednak zaprzeczyć, że należy dobrze znać swój komputer, aby móc prawidłowo skonfigurować jądro. Większość informacji można zdobyć poprzez instalację pakietu pciutils (emerge pciutils) zawierającego program lspci. Dzięki temu będzie możliwe używanie lspci wewnętrz chrootowanego środowiska. Podczas pracy z tym programem można bezpiecznie zignorować wszelkie ostrzeżenia związane z pcilib (jak np. "pcilib: cannot open /sys/bus/pci/devices"). Ponadto można również uruchomić lspci poza środowiskiem chroot. Powinno dać to taki sam efekt. Dodatkowe informacje o sterownikach, które należy włączyć do jądra można uzyskać dzięki poleceniu lsmod, które pokażemy listę modułów jakie załadował system płyty instalacyjnej.

Kiedy już zbierzemy wszystkie informacje przechodzimy do katalogu ze źródłami i wpisujemy polecenie make menuconfig. Uruchomimy się menu konfiguracyjne oparte na bibliotekach graficznych ncurses.

Listing 4: Wywoływanie menu konfiguracyjnego

```
# cd /usr/src/linux
# make menuconfig
```

Cała konfiguracja została podzielona na kilka sekcji, co znacznie ułatwia odnalezienie i wybranie odpowiednich sterowników. Poniżej wymieniamy wszystkie opcje, które należy włączyć, aby Gentoo mogło prawidłowo funkcjonować.

Zaznaczanie wymaganych ustawień

Przede wszystkim należy zapewnić sobie możliwość korzystania z rozwojowych i eksperymentalnych fragmentów kodu jądra. Jeśli się z tego zrezygnuje to zniknie kilka bardzo ważnych ustawień.

Listing 5: Wybieranie opcji experimental code/drivers

```
Code maturity level options --->
[*] Prompt for development and/or incomplete code/drivers
```

Wszystkie sterowniki potrzebne do wystartowania systemu (takie jak kontroler SCSI) muszą być wkompilowane na stałe w jądro - nie należy ich dodawać w formie modułów.

Na samym początku należy wybrać odpowiednią dla posiadanej sprzętowej rodziny procesorów:

Listing 6: Wybór rodziny procesorów

```
Processor type and features --->
(Należy zmienić wybór w zależności od konfiguracji sprzętowej komputera)
(Athlon/Duron/K7) Processor family
```

Następnie przechodzimy do File Systems i wybieramy obsługę systemów plików, z których zamierzamy używać. Jeśli pominiemy ten krok, to Gentoo nie będzie w stanie zamontować niektórych partycji lub nawet się nie uruchomi. Sterowniki te również powinny być wkompilowane w jądro na stałe, nie należy ich dodawać w modułach. Oprócz tego zaznaczamy opcje Virtual memory i /proc file system. Użytkownicy jąder 2.4 wciąż muszą korzystać z /dev file system, ponieważ jądra te nie obsługują udev.

Jeśli korzysta się z jądra 2.4, należy wybrać /dev file system w związku z tym, że jądra z serii 2.4 nie posiadają obsługi udev.

Listing 7: Wybór potrzebnych systemów plików

```
(Dla jądra 2.4)
File systems --->
[*] Virtual memory file system support (former shm fs)
[*] /proc file system support
[*] /dev file system support (EXPERIMENTAL)
[*] automatically mount /dev at boot
[ ] /dev/pts file system for Unix98 PTYs
```

```
(Dla jądra 2.6)
File systems --->
Pseudo Filesystems --->
[*] /proc file system support
[*] Virtual memory file system support (former shm fs)
```

```
(Zaznaczamy opcje odpowiednie dla używanych systemów plików)
<*> Reiserfs support
<*> Ext3 journaling file system support
<*> JFS filesystem support
<*> Second extended fs support
<*> XFS filesystem support
```

Jeśli BIOS nie radzi sobie z dużymi dyskami i konieczne było ograniczenie maksymalnego ich rozmiaru przy pomocy jumpera warto zaznaczyć poniższą opcję, co umożliwi uzyskanie dostępu również do niewidocznej dla BIOS-u części urządzenia.

Listing 8: Zaznaczanie opcji Auto-Geometry Resizing

```
(Wyłącznie dla jąder 2.4.X)
ATA/IDE/MFM/RLL support --->
  IDE, ATA and ATAPI Block devices --->
    <*>   Include IDE/ATA-2 DISK support
    [ ]     Use multi-mode by default
    [*]     Auto-Geometry Resizing support
```

Nie zapominajmy o włączeniu DMA dla dysków:

Listing 9: Włączanie obsługi DMA

```
Device Drivers --->
  ATA/ATAPI/MFM/RLL support --->
    [*] Generic PCI bus-master DMA support
    [*] Use PCI DMA by default when available
```

Użytkownicy łączący się z Internetem za pomocą PPPoE i połączeń dial-up powinni zaznaczyć również:

Listing 10: Sterowniki niezbędne dla użytkowników PPPoE

```
Device Drivers --->
  Networking Support --->
    <*> PPP (point-to-point protocol) support
    <*> PPP support for async serial ports
    <*> PPP support for sync tty ports
```

Listing 11: Sterowniki niezbędne dla użytkowników PPPoE cd.

```
(Dla jąder 2.4)
Network device support --->
  <*> PPP (point-to-point protocol) support
  <*> PPP support for async serial ports
  <*> PPP support for sync tty ports
```

```
(Dla jąder 2.6)
Device Drivers --->
  Networking support --->
    <*> PPP (point-to-point protocol) support
    <*> PPP support for async serial ports
    <*> PPP support for sync tty ports
```

Obie opcje dotyczące kompresji nie są wprawdzie wymagane, ale również nie zaszkodzą naszemu systemowi, podobnie zresztą jak opcja PPP over Ethernet, która jest przydatna tylko gdy skonfiguruje się rp-pppoe do pracy w trybie PPPoE jądra (kernel mode PPPoE).

Należy wkompilować odpowiednie sterowniki dla wszystkich posiadanych kart sieciowych.

Posiadacze procesorów Intel z technologią HyperThreading (tm) lub systemów wieloprocesorowych powinni zaznaczyć opcje "Symmetric multi-processing support":

Listing 12: Włączanie SMP

```
Processor type and features --->
 [*] Symmetric multi-processing support
```

Uwaga: W systemach wielordzeniowych, każdy z rdzeni widziany jest jako osobny procesor.

Jeśli posiadamy urządzenia wejściowe USB (np. klawiaturę lub myszkę) dodajmy sterowniki również dla nich:

Listing 13: Aktywowanie wsparcia dla urządzeń wejścia używających USB

```
Device Drivers --->
 USB Support --->
 <*>   USB Human Interface Device (full HID) support
```

Posiadacze laptopów chcący korzystać z PCMCIA nie powinni korzystać ze sterowników PCMCIA z jądra 2.4. Za chwilę omówimy instalację nowszych sterowników z pakietu pcmcia-cs. Użytkownicy jąder 2.6 mogą bezpiecznie korzystać ze sterowników z jądra.

Poza włączeniem generalnej obsługi PCMCIA w jądrze 2.6, należy również zaznaczyć używany przez karty PCMCIA mostek obecny w systemie:

Listing 14: Włączanie obsługi PCMCIA w jądrach 2.6

```
Bus options (PCI, PCMCIA, EISA, MCA, ISA) --->
 PCCARD (PCMCIA/CardBus) support --->
 <*> PCCard (PCMCIA/CardBus) support
(Starsze karty wymagają tu opcji 16 bit - zapewne większość użytkowników zechce włączyć ta opcję)
 <*> 16-bit PCMCIA support
 [*] 32-bit CardBus support
(Wybieramy odpowiednie mostki)
 --- PC-card bridges
 <*> CardBus yenta-compatible bridge support (NEW)
 <*> Cirrus PD6729 compatible bridge support (NEW)
 <*> i82092 compatible bridge support (NEW)
 <*> i82365 compatible bridge support (NEW)
 <*> Databook TCIC host bridge support (NEW)
```

Po skonfigurowaniu kernela przyszła pora na jego skompilowanie i instalację. Opuszczamy program konfiguracyjny i rozpoczynamy proces komplikacji:

Listing 15: Kompilowanie kernela

```
(Dla jąder 2.4)
# make dep && make bzImage modules modules_install

(Dla jąder 2.6)
# make && make modules_install
```

Kiedy jądro skończy się komplikować należy przekopiować jego obraz do katalogu /boot. Wybieramy dowolną nazwę dla naszego jądra jednak należy pamiętać jaka to nazwa, gdyż będziemy jej potrzebowali w późniejszym czasie w trakcie konfiguracji bootloadera. Należy pamiętać o zastąpieniu części wpisu kernel-2.6.17-gentoo-r5 swoją nazwą i wersją kernela.

Listing 16: Instalowanie jądra

```
# cp arch/i386/boot/bzImage /boot/kernel-2.6.17-gentoo-r5
```

Następnie przechodzimy do akapitu dotyczącego modułów.

7.d. Alternatywnie: użycie genkernela

Ten paragraf jest przeznaczony dla użytkowników, którzy zdecydowali się użyć programu genkernel do skonfigurowania jądra.

Po zainstalowaniu źródeł należy je skonfigurować. Zrobimy to automatycznie przy pomocy programu genkernel, który wykonuje cały proces dokładnie w ten sam sposób w jaki jest konfigurowane jądro na płycie instalacyjnej. Konsekwencją wyboru genkernela jest to, że system będzie zmuszony do wykrywania dostępnego sprzętu przy każdym uruchomieniu komputera. W związku z tym, że genkernel nie wymaga od użytkownika żadnych ręcznych poprawek w konfiguracji, jest doskonałym rozwiązaniem dla tych wszystkich, którzy nie są najmocniejsi w samodzielnym kompilowaniu jądra.

Zanim jednak zdradzimy jak używa się tego cudownego programu musimy wytlumaczyć jak go zainstalować:

Listing 17: Instalowanie genkernela

```
# emerge genkernel
```

Użytkownicy jąder serii 2.6 mogą od razu skopiować konfigurację z płyty instalacyjnej do miejsca skąd będzie mógł odczytać ją genkernel.

Listing 18: Kopiowanie konfiguracji genkernela z płyty instalacyjnej

(Wyłącznie dla użytkowników jąder serii 2.6)

```
# zcat /proc/config.gz > /usr/share/genkernel/x86/kernel-config-2.6
```

Źródła skompilujemy przy pomocy polecenia genkernel all. Kompilowanie zajmie mnóstwo czasu, ponieważ genkernel zawiera niemal wszystkie dostępne sterowniki.

Jeśli na partycji rozruchowej został stworzony inny system plików niż ext2 lub ext3 to konieczne będzie dodanie potrzebnych sterowników, które można wybrać za pomocą menu wywołanego poleceniem genkernel --menuconfig all. Sterowniki te muszą być wkompilowane w jądro na stałe, nie można dodawać ich w postaci modułów. Użytkownicy EVMS2 lub LVM2 powinni dodać również --evms2.

Listing 19: Uruchamianie genkernela

```
# genkernel all
```

W toku tego procesu powstanie właściwy plik jądra, initrd (initial root disk) oraz ogromna rzesza modułów. Nazwy plików jądra i initrd będą potrzebne przy konfiguracji bootloadera do prawidłowego wypełnienia jego pliku konfiguracyjnego, więc warto je sobie zapisać. Przy następnym uruchomieniu komputera zostanie najpierw wykonany plik initrd, który wykryje cały dostępny sprzęt i wczyta odpowiednie moduły, a następnie uruchomi się właściwy system.

Listing 20: Sprawdzanie nazw utworzonych plików jądra

```
# ls /boot/kernel* /boot/initramfs*
```

Następna czynność jeszcze bardziej upodobni nasz system do tego dostępnego na płycie instalacyjnej, zemerujemy coldplug. Podczas gdy initrd automatycznie wykrywa sprzęt, co jest niezbędne do uruchomienia systemu, coldplug będzie wykrywał wszystko inne. Aby zemerować i uaktywnić coldplug należy wpisać:

Listing 21: Instalacja i uruchamianie coldplug

```
# emerge coldplug
# rc-update add coldplug boot
```

7.e. Moduły jądra

Konfigurowanie modułów

Uwaga: Jeśli korzysta się z jądra 2.4, należy zastąpić wszystkie 2.6 w poniższym tekście i poleceniach przez 2.4.

Moduły ładowane w czasie startu systemu muszą zostać dopisane do pliku /etc/modules.autoload.d/kernel-2.6. Można tu również dodać dodatkowe opcje ich ładowania.

Listę wszystkich dostępnych modułów pokaże polecenia find z podaną poniżej składnią. Należy zastąpić wpis 2.6.17-r5 wersją jądra, które zostało właśnie zbudowane.

Listing 22: Znajdowanie dostępnych modułów

```
# find /lib/modules/2.6.17-r5/ -type f -iname '*.o' -or -iname '*.ko'
```

Dla przykładu, aby załadować moduł 3c59x.o należy edytować plik kernel-2.4 lub kernel-2.6 i dodać tam jego nazwę.

Listing 23: Edytowanie /etc/modules.autoload.d/kernel-2.4

```
# nano -w /etc/modules.autoload.d/kernel-2.6
```

Listing 24: /etc/modules.autoload.d/kernel-2.6

```
3c59x
```

8. Konfigurowanie systemu

8.a. Informacje o systemach plików

Co to jest fstab?

W Linuksie wszystkie używane przez system partycje powinny być wpisane do /etc/fstab. Plik ten zawiera informacje o tym gdzie w strukturze katalogów), z jakimi opcjami i kiedy (automatycznie przy starcie systemu, czy nie, przez zwykłych użytkowników, czy nie itd.) mają zostać zamontowane.

Tworzenie /etc/fstab

Plik /etc/fstab używa specyficznej składni. Wszystkie wiersze składają się z sześciu pól, oddzielonych spacjami lub/i tabulatorami. Każde z nich pełni określoną funkcję:

- Pierwsze pole definiuje **partycję** (ścieżkę do odpowiadającego jej urządzenia).
- Drugie pole kontroluje **punkt montowania**.
- Trzecie pole opisuje używany przez partycję **system plików**.
- W czwartym polu podane są **opcje montowania** używane przez mount. Każdy system plików posiada własne ustawienia, pełna lista znajduje się w podręczniku systemowym programu mount (man mount). Wszystkie opcje powinny być oddzielone przecinkami.
- Piąte pole używane jest przez dump do ustalenia czy dana partycja ma być **dumpowana** czy nie. Zazwyczaj należy wpisać tu 0 (zero).
- Z szóstego pola korzysta fsck do ustalenia kolejności **sprawdzania** partycji po nieprawidłowym wyłączeniu systemu. Dla głównego systemu plików należy wpisać 1, natomiast dla pozostałych 2 (lub 0 jeśli kontrola nie jest konieczna).

Ważne: Domyślny /etc/fstab dostarczany przez Gentoo nie jest poprawnym plikiem fstab, uruchamiamy więc nano (lub inny ulubiony edytor) i tworzymy własny plik /etc/fstab:

Listing 1: Tworzenie /etc/fstab

```
# nano -w /etc/fstab
```

Spójrzmy jak zapisać opcje partycji /boot. To tylko przykład, jeśli nie korzysta się z partycji rozruchowej (/boot), nie należy go kopować.

W naszym przykładowym schemacie (dla x86) /boot będzie partycją /dev/hda1 (lub /dev/sda* dla SATA i SCSI) i będzie używał systemu plików ext2 oraz będzie sprawdzany podczas rozruchu.

Listing 2: Przykładowy wpis do /etc/fstab dla /boot

```
/dev/hda1      /boot      ext2      defaults          1  2
```

Niektórzy użytkownicy ze względów bezpieczeństwa nie chcą, aby partycja /boot była montowana automatycznie. Powinni oni zastąpić opcję defaults opcją noauto. Potem trzeba będzie ręcznie zamontować tą partycję przed każdym jej użyciem.

Dodajemy regułki, które odpowiadają naszemu schematowi podziału oraz linie dla /proc, tmpfs, napędu CD-ROM i innych dysków, jeśli są zainstalowane w komputerze.

Następnie używamy poniższego przykładu do stworzenia pliku /etc/fstab:

Listing 3: Plik /etc/fstab

```
/dev/hda1      /boot      ext2      defaults,noatime   1  2
/dev/hda2      none       swap      sw                  0  0
/dev/hda3      /          ext3      noatime            0  1

none          /proc       proc      defaults            0  0
none          /dev/shm    tmpfs     nodev,nosuid,noexec 0  0

/dev/cdrom    /mnt/cdrom auto      noauto,user        0  0
```

Opcja auto powoduje, że mount sam próbuje wykryć system plików (zalecane dla wymienialnych nośników, które mogą posiadać różne systemy), a user umożliwia montowanie zwykłym użytkownikom.

Aby zwiększyć wydajność, należy dodać opcję noatime do parametrów montowania. Dzięki temu można skrócić czas dostępu do partycji i znacznie przyspieszyć system. Opcja ta powoduje, że czasy dostępu nie będą zapisywane, a ta informacja nie jest do niczego potrzebna znakomitej większości użytkowników.

Sprawdzamy ponownie /etc/fstab, zapisujemy zmiany i zamykamy plik.

8.b. Konfiguracja sieci

Nazwa hosta

Każdy użytkownik powinien nadać swojemu komputerowi jakąś nazwę. Wydaje się to proste, ale wielu ma z tym spore trudności. Zawsze można tę nazwę zmienić. My wybraliśmy host tux oraz domenę homenetwork.

Listing 4: Konfiguracja nazwy hosta

```
# nano -w /etc/conf.d/hostname
(Ustawienie zmiennej HOSTNAME)
HOSTNAME="tux"
```

Konfiguracja sieci

Zanim powiemy "Hej, przecież już to zrobiliśmy!" należy pamiętać, że to co ustawialiśmy na początku instalacji jest przeznaczone tylko na jej potrzeby. Teraz ostatecznie skonfigurujemy sieć dla instalowanego systemu Gentoo.

Uwaga: Szczegółowe informacje dotyczące zagadnień sieciowych, takich jak bonding, bridging, VLAN czy 802.11q, znajdują się w rozdziale dotyczącym Konfiguracji sieci.

Wszystkie ustawienia dotyczące sieci znajdują się w /etc/conf.d/net. Mają prostą, ale niekoniecznie intuicyjną składnię. Nie ma czego się obawiać, wszystko zostanie wyjaśnione. Warto zapoznać się z przykładowym plikiem /etc/conf.d/net.example, w którym znajduje się wiele cennych wskazówek oraz kilka przykładowych konfiguracji sieci.

Domyślnym ustawieniem jest DHCP, dlatego jego użytkownicy nie muszą dokonywać w plikach żadnych zmian. Nie zwalnia to ich jednak z konieczności zainstalowania klienta DHCP. Wszystko na ten temat znajduje się w rozdziale Instalowanie narzędzi systemowych.

Jeśli jednak zajdzie potrzeba dokonfigurowania sieci, np. by wybrać określone opcje dla DHCP lub całkowicie zrezygnować z jego użycia, należy otworzyć plik /etc/conf.d/net w ulubionym edytorze (w przykładzie użyjemy nano):

Listing 5: Otwieranie /etc/conf.d/net do edycji

```
# nano -w /etc/conf.d/net
```

Znajduje się tam następujący wpis:

Listing 6: Domyślny /etc/conf.d/net

```
config_eth0=( "dhcp" )
# This blank configuration will automatically use DHCP for any net.*
# scripts in /etc/init.d. To create a more complete configuration,
# please review /etc/conf.d/net.example and save your configuration
# in /etc/conf.d/net (this file :]).
```

Gdy IP, maska sieciowa oraz brama są ustawiane ręcznie to edytujemy obie zmienne, config_eth i routes_eth0:

Listing 7: Ręczne ustawianie informacji o IP dla eth0

```
config_eth0=( "192.168.0.2 netmask 255.255.255.0 brd 192.168.0.255" )
routes_eth0=( "default gw 192.168.0.1" )
```

Aby wybrać określone opcje DHCP należy dodać zmienne config_eth0 i dhcp_eth0:

Listing 8: Automatyczne pobieranie adresu IP dla eth0

```
config_eth0=( "dhcp" )
dhcp_eth0="nodns nontp nonis"
```

Powtarzamy powyższe instrukcje dla pozostałych interfejsów sieciowych (odpowiednio config_eth1, config_eth2).

Lista dostępnych ustawień znajduje się w pliku /etc/conf.d/net.example.

Następnie należy zapisać konfigurację i zamknąć edytor.

Automatyczny start sieci podczas uruchamiania systemu

Aby urządzenia sieciowe były aktywowane podczas startu, musimy je dodać do domyślnego poziomu uruchamiania. Posiadacze urządzeń PCMCIA mogą pominąć te czynności, gdyż są one inicjowane przez osobny skrypt startowy.

Listing 9: Dodawanie net.eth0 do domyślnego poziomu uruchamiania

```
# rc-update add net.eth0 default
```

Posiadacze kilku urządzeń sieciowych muszą utworzyć odpowiednie skrypty startowe, np. net.eth1, net.eth2 itd. Można w tym celu skorzystać z ln:

Listing 10: Tworzenie dodatkowych skryptów startowych

```
# cd /etc/init.d
# ln -s net.lo net.eth1
# rc-update add net.eth1 default
```

Zapisywanie informacji o sieci

Trzeba poinformować system o istnieniu lokalnej sieci. Służy do tego plik /etc/hosts. Zapisujemy w nim nazwy hostów i odpowiadające im adresy IP, których nie może ustalić serwer nazw. Będziemy musieli w tym pliku zdefiniować nasz komputer. Dodatkowo, możemy tutaj również umieścić komputery z naszej sieci jeżeli nie będziemy chcieli konfigurować wewnętrznego serwera DNS.

Listing 11: Otwieranie /etc/hosts

```
# nano -w /etc/hosts
```

Listing 12: Wpisywanie informacji o sieci

```
(Wpis definiujący nasz komputer) .
127.0.0.1      localhost

(Definiujemy pozostałe komputery z naszej sieci. Muszą one posiadać IP przypisane na stałe, aby skorzystać z tego sposobu).
192.168.0.5    jenny.homenetwork jenny
192.168.0.6    benny.homenetwork benny
```

Zapisujemy zmiany i zamykamy edytor.

Osoby nie posiadające PCMCIA mogą od razu przejść do sekcji Konfiguracja systemu. W przeciwnym wypadku należy czytać dalej.

Opcjonalnie: Konfiguracja PCMCIA

Posiadacze PCMCIA muszą zainstalować pakiet pcmcia-cs. Dotyczy to także użytkowników jąder z serii 2.6 (nawet jeżeli nie będą używali sterowników PCMCIA z tego pakietu).

Aby pominąć opcjonalną zależność xorg-x11, wpisujemy dodatkowo USE="-X":

Listing 13: Instalacja pcmcia-cs

```
# USE="-X" emerge pcmcia-cs
```

Następnie dodajemy do domyślnego poziomu uruchamiania skrypt pcmcia:

Listing 14: Dodawanie do domyślnego poziomu uruchamiania skryptu pcmcia

```
# rc-update add pcmcia default
```

8.c. Konfiguracja systemu

Hasło superużytkownika

Hasło roota zmieniamy poleceniem:

Listing 15: Ustawienie hasła superużytkownika

```
# passwd
```

Jeżeli chcemy, aby superużytkownik mógł logować się przez konsolę szeregową, dodajemy tts/0 do /etc/securetty:

Listing 16: Dodawanie tts/0 do /etc/securetty

```
# echo "tts/0" >> /etc/securetty
```

Informacje o systemie

Do najbardziej podstawowych ustawień Gentoo używa pliku /etc/rc.conf. Otwieramy go i zapoznajemy się z umieszczonymi w nim komentarzami. :)

Listing 17: Otwieranie /etc/rc.conf

```
# nano -w /etc/rc.conf
```

Po dokonaniu zmian należy zapisać je do pliku.

Jak widać, plik ten jest dobrze skomentowany. Dzięki temu można poradzić sobie z umieszczonymi w nim zmiennymi bez niemal żadnych problemów. Między innymi można tu skonfigurować czcionki używane przez system i menedżer uruchamiania serwera X (jak kdm czy gdm).

Konfiguracja klawiatury znajduje się w pliku /etc/conf.d/keymaps i to jego należy edytować w celu zmiany ustawień.

Listing 18: Otwieranie /etc/conf.d/keymaps

```
# nano -w /etc/conf.d/keymaps
```

Zmienna KEYMAP wymaga specjalnego traktowania. Jeśli zostanie wybrana zła wartość to mogą pojawić się dziwne rezultaty podczas pisania na klawiaturze.

Po dokonaniu zmian należy zapisać plik i opuścić edytor.

Ustawienia zegara w Gentoo znajdują się w pliku /etc/conf.d/clock. Należy go wyedytować i poprawić ustawienia.

Listing 19: Otwieranie /etc/conf.d/clock

```
# nano -w /etc/conf.d/clock
```

Jeśli zegar sprzętu jest inny niż UTC należy dodać do pliku opcję CLOCK="local", aby godzina w systemie zgadzała się z rzeczywistością.

Po ukończeniu edycji zapisujemy zmiany i zamykamy edytor.

9. Instalowanie narzędzi systemowych

9.a. Menedżer urządzeń

Użytkownicy jąder 2.4 instalujący Gentoo z etapu trzeciego muszą na początek odinstalować udev i zainstalować devfsd. Jest to niezbędne w związku z tym, że Gentoo domyślnie korzysta z systemu udev niedostępnego na jądrach 2.4.

Listing 1: Instalowane devfsd

```
(Dla użytkowników jąder 2.4.x używających stage 3)
# emerge --unmerge udev
# emerge devfsd
```

9.b. Program logujący

W archiwum stage3 brakuje kilka ważnych programów, gdyż kilka pakietów spełnia te same funkcje, a my nie chcemy dokonywać ich wyboru w imieniu użytkownika.

Pierwszym narzędziem przy którym należy dokonać wyboru, jest program do obsługi systemu logowania. Unix i Linux posiadają bogatą historię w tym zakresie. Jeśli to konieczne, można logować do plików wszystko, co dzieje się w systemie. Mechanizmem tym zarządza właśnie program logujący.

Gentoo oferuje kilka różnych programów logujących: sysklogd - tradycyjny zestaw logujących demonów, syslog-ng - zaawansowany program logujący oraz metalog charakteryzujący się dużą liczbą opcji konfiguracyjnych. W Portage znajduje cały wachlarz programów logujących i nie tylko - liczba naszych pakietów rośnie z każdym dniem.

Jeżeli planuje się używanie sysklogd lub syslog-ng dobrym pomysłem jest zainstalowanie programu logrotate, ponieważ te programy logujące nie są zaopatrzone w żaden mechanizm rotacyjny dla logów.

Aby zainstalować wybrany program logujący, korzystamy z polecenia emerge, a następnie dodajemy go do domyślnego poziomu startowego poprzez skrypt rc-update. Poniższy przykład przedstawia proces instalacji programu syslog-ng:

Listing 2: Instalacja programu logującego

```
# emerge syslog-ng
# rc-update add syslog-ng default
```

9.c. Opcjonalnie: Demon Cron

Następnym programem jest demon Cron. Pomimo, że jest on opcjonalny i nie jest wymagany do poprawnej pracy systemu, zalecane jest jego zainstalowanie. Czym jest demon Cron? Jest to program służący do wykonywania zaplanowanych poleceń w określonym czasie. Jest on bardzo przydatny, gdy wykonujemy pewne czynności regularnie (na przykład codziennie, co tydzień, co miesiąc).

Gentoo oferuje trzy różne demony crona: dcron, fcron oraz vixie-cron. Instalacja każdego z nich jest analogiczna do instalacji programu logującego, jednakże dcron i fcron wymagają dodatkowej konfiguracji (wykonywanej przez polecenie: crontab /etc/crontab). Niezdecydowanym polecamy program vixie-cron.

Dla instalacji bez sieci dostarczamy tylko vixie-cron. Aby używać innego demona cron, trzeba będzie poczekać i zainstalować go później.

Listing 3: Instalacja demona cron

```
# emerge vixie-cron
# rc-update add vixie-cron default
(Dla dcron lub fcron) # crontab /etc/crontab
```

9.d. Opcjonalnie: Indeksowanie plików

Aby możliwe było indeksowanie plików w systemie w celu ich szybkiego wyszukiwania za pomocą narzędzia locate, należy zainstalować pakiet sys-apps/slocate.

Listing 4: Instalacja slocate

```
# emerge slocate
```


9.e. Narzędzia obsługi systemu plików

W zależności od tego, jakiego systemu plików używamy, musimy zainstalować odpowiednie narzędzia do jego obsługi (do sprawdzania jego integralności, czy tworzenia dodatkowych systemów plików).

W poniższej tabeli przedstawiono narzędzia, których należy użyć dla poszczególnych używanych systemów plików:

System plików	Narzędzie	Polecenie instalujące
XFS	xsprogs	emerge xsprogs
ReiserFS	reiserfsprogs	emerge reiserfsprogs
JFS	jfsutils	emerge jfsutils

Użytkownicy EVMS powinni zainstalować pakiet evms:

Listing 5: Instalacja narzędzi EVMS

```
# USE="-gtk" emerge evms
```

Parametr USE="-gtk" spowoduje, że nie zostaną zainstalowane wszystkie zależności. Aby w przyszłości skorzystać z graficznych nakładek na evms należy przebudować ten pakiet bez tej flagi.

Jeżeli nie potrzeba żadnych dodatkowych narzędzi sieciowych (takich jak rp-pppoe lub klient dhcp) przechodzimy do Konfiguracji bootloadera.

9.f. Narzędzia sieciowe

Opcjonalnie: Instalowanie klienta DHCP

Jeżeli chcemy, aby Gentoo automatycznie uzyskiwało adres IP karty sieciowej, musimy zainstalować dhcpcd (lub jakiegokolwiek innego klienta DHCP - opis w Modularna praca w sieci). Jeżeli nie zrobi się tego teraz, połączenie sieciowe może nie działać po zakończeniu instalacji!

Listing 6: Instalacja dhcpcd

```
# emerge dhcpcd
```

Opcjonalnie: Instalacja klienta PPPoE

Jeśli do łączenia się z siecią potrzeba nam rp-pppoe, należy zainstalować wymagane narzędzia.

Listing 7: Instalacja klienta PPPoE

```
# USE="-X" emerge rp-pppoe
```

Ustawienie USE="-X" zapobiegnie instalacji pakietu xorg-x11 poprzez zależności (rp-pppoe posiada narzędzia graficzne, aby z nich korzystać, można przekompilować rp-pppoe po instalacji serwera xorg-x11 lub zainstalować go teraz, jednak jego komplikacja zajmie dużo czasu).

Opcjonalnie: Narzędzia RAID dla sprzętu IBM

Użytkownicy SCSI RAID na systemach opartych na POWER-5 powinni rozważyć instalację pakietu iprutils, dzięki któremu możliwe będą operacje na macierzach dysków RAID, takie jak pobieranie ich statusu oraz uaktualnianie mikrokodu.

Listing 8: Instalowanie iprutils

```
# emerge iprutils
```

10. Konfiguracja bootloadera

10.a. Podejmowanie decyzji

Po skonfigurowaniu jądra i modyfikacji odpowiednich plików konfiguracyjnych systemu, można przejść do etapu instalacji programu, który uruchomi jądro w momencie uruchomienia systemu. Taki program nazywa się bootloader.

Dla architektury x86, Gentoo Linux używa dwóch takich programów: [GRUB](#) oraz [LILO](#).

Przed instalacją któregokolwiek z nich opiszemy jeszcze jak skonfigurować bufor ramki, umożliwiający poprawę wyglądu konsoli. Jeśli użytkownik nie chce korzystać z framebuffera, może pominąć rozdział o nim.

Opcjonalnie: Bufor ramki

Jeżeli jądro zostało skompilowane z obsługą bufora ramki (lub użyto genkernela), należy dodać parametr vga i/lub video w odpowiednim miejscu pliku konfiguracyjnego bootloadera, aby go aktywować.

Najważniejszą informacją jaką należy poznać przed rozpoczęciem konfiguracji jest nazwa używanego urządzenia bufora ramki. Niektóre jądra, takie jak na przykład gentoo-sources, domyślnie posiadają obsługę vesafb-tng jako sterownika typu VESA. Użytkownicy vesafb-tng nie muszą dodawać parametru vga. Pozostali, używający vesafb, wciąż muszą dopisywać vga.

Za pomocą parametru vga określa się rozdzielcość oraz głębię kolorów ekranu bufora ramki vesafb. Każdy numer określa inną głębię i rozdzielcość. Ich lista znajduje się w dokumencie /usr/src/linux/Documentation/fb/vesafb.txt, który instaluje się w systemie wraz ze źródłami jądra.

Oto tabela parametrów vga, która zawiera wszystkie dostępne rozdzielcości i głębie kolorów:

640x480 800x600 1024x768 1280x1024				
256	0x301	0x303	0x305	0x307
32k	0x310	0x313	0x316	0x319
64k	0x311	0x314	0x317	0x31A
16M	0x312	0x315	0x318	0x31B

Parametr video odpowiada za opcje wyświetlania bufora ramki. Jest mu niezbędna nazwa sterownika bufora ramki (vesafb dla jader 2.6 lub vesa w 2.4) oraz opcje z jakimi ma być uruchamiany bufor. Wszystkie zmienne znajdują się w pliku /usr/src/linux/Documentation/fb/vesafb.txt, oto najważniejsze z nich:

Zmienna Opis

ywrap	Włącza zawijanie pamięci karty graficznej
mtrr:n	n może być: 0 - disabled 1 - uncachable 2 - write-back 3 - write-combining 4 - write-through (tylko dla vesafb-tng)
mode	Pozwala na ustawienie rozdzielcości, głębi kolorów oraz częstotliwości odświeżania. Na przykład 1024x768-32@85 ustawia rozdzielcość 1024x768, 32-bitową głębię kolorów i częstotliwość odświeżania 85 Hz.

Wynik tych działań powinien wyglądać mniej więcej tak: vga=0x318 video=vesafb:mtrr:3,ywrap lub video=vesafb:mtrr,ywrap,1024x768-32@85. Warto zapamiętać te ustawienia, wkrótce będą potrzebne.

Kolejny etap to instalacja GRUB lub LILO.

10.b. Domyślnie: Użycie programu GRUB

Wyjaśnienie terminologii związanej z GRUB

Najważniejszym czynnikiem prowadzącym do zrozumienia programu GRUB jest pojęcie jego sposobu odnoszenia się do dysków i partycji. Partycja /dev/hda1 (dla dysków IDE) lub /dev/sda1 (dla dysków SCSI/SATA) jest określana przez GRUB-a jako (hd0,0). Zauważmy nawiasy po obu stronach hd0,0 - ich użycie jest wymagane.

Dyski twarde są liczone od zera, a nie od "a". Partycje także zaczynają się od zera, a nie od jedynki. W grupie urządzeń pamięci masowej tylko twarde dyski są liczone, nie ma to natomiast miejsca w przypadku napędów ATAPI-IDE takich jak CD-ROM-y i nagrywarki. Ta sama sytuacja ma miejsce w przypadku dysków SCSI. (Normalnie otrzymują one wyższe numery niż dyski IDE, z wyjątkiem sytuacji, gdy BIOS jest ustawiony na start systemu z urządzenia SCSI).

Zakładając, że w komputerze jest zainstalowany dysk twarty oznaczony jako /dev/hda, odtwarzacz cdrom /dev/hdb, nagrywarka /dev/hdc, drugi dysk twarty /dev/hdd i nie ma żadnego dysku SCSI, urządzenie /dev/hdd7 jest rozumiane jako (hd1,6). Całość wygląda dość pokrętnie, jednakże GRUB oferuje mechanizm uzupełniania wiersza polecenia klawiszem TAB, co z pewnością jest wybawieniem dla tych, którzy mają dużo dysków twardych i partycji, a gubią się nieco w sposobie numeracji używanym przez program GRUB.

Nadeszła pora na zainstalowanie GRUB-a.

Instalacja programu GRUB

Po pierwsze należy zainstalować program GRUB:

Listing 1: Instalowanie GRUB-a

```
# emerge grub
```

GRUB został zainstalowany, teraz należy go jeszcze skonfigurować i umieścić w MBR, tak by automatycznie uruchamiał nowo zainstalowane jądra. Najpierw przy pomocy nano lub dowolnego innego edytora należy utworzyć plik /boot/grub/grub.conf:

Listing 2: Tworzenie /boot/grub/grub.conf

```
# nano -w /boot/grub/grub.conf
```

Następnie należy dokonać edycji pliku grub.conf. Poniżej znajdują się dwa przykłady plików grub.conf, odpowiadające schematowi partycjonowania użytkemu w tym przewodniku. Tylko pierwszy z nich został wyczerpująco opatrzony komentarzami.

- Pierwszy grub.conf jest dla osób, które nie wykorzystały programu genkernel do budowania jądra
- Drugi grub.conf jest dla tych, którzy użyli programu genkernel do zbudowania jądra.

Uwaga: Jeśli system plików głównej partycji to JFS, należy dodać "ro" do linii kernel w związku z tym, że JFS musi odtworzyć swój log zanim pozwoli na montowanie w trybie do odczytu i zapisu.

Listing 3: grub.conf dla osób nie używających genkernel

```
# Która pozycja ma być uruchamiana domyślnie. 0 oznacza pierwszą, 1 drugą itd.
default 0
# Ile sekund oczekiwania przed uruchomieniem pozycji domyślnej.

timeout 30
# Ładny obrazek na zaostرzenie apetytu
# Dla użytkowników, którzy skonfigurowali bufor ramki
splashimage=(hd0,0)/boot/grub/splash.xpm.gz

title=Gentoo Linux 2.6.17-r5
# Partycja na której znajduje się obraz jądra (lub system operacyjny)

root (hd0,0)
kernel /boot/kernel-2.6.17-gentoo-r5 root=/dev/hda3

# Następne trzy linie są potrzebne tylko gdy będzie uruchamiany również system Windows.
# W tym przypadku Windows znajduje się na /dev/hda6.
title=Windows XP
rootnoverify (hd0,5)
makeactive
chainloader +1
```

Listing 4: grub.conf dla użytkowników genkernela

```
default 0
timeout 30
splashimage=(hd0,0)/boot/grub/splash.xpm.gz

title=Gentoo Linux 2.6.17-r5
root (hd0,0)
kernel /boot/kernel-genkernel-x86-2.6.17-gentoo-r5 root=/dev/ram0 init=/linuxrc
ramdisk=8192 real_root=/dev/hda3 udev
initrd /boot/initramfs-genkernel-x86-2.6.17-gentoo-r5
```

```
# Tylko, gdy będą uruchamiane dwa systemy
title=Windows XP
rootnoverify (hd0,5)
makeactive
chainloader +1
```

Uwaga: Wspomniany na końcu linii append udev został dodany w celu obejścia jednego z błędów genkernela, który czasem nie współpracuje poprawnie z udev.

Jeśli użyto innego schematu partycjonowania lub/i obrazu jądra, należy nanieść odpowiednie poprawki na konfigurację. Należy się wtedy upewnić, że wszystko co się dopisuje w konfiguracji co jest dopisywane w formacie nazewnictwa urządzeń specyficznych dla GRUB-a (w stylu (hd0,0)) jest względne względem punktu montowania, a nie katalogu głównego. Innymi słowy, (hd0,0)/grub/splash.xpm.gz powinien być dopisany jako /boot/grub/splash.xpm.gz w związku z tym, że (hd0,0) to /boot.

Jeśli wybrano inny schemat partycjonowania, a /boot nie znajduje się na osobnej partycji, prefiks /boot użyty w powyższych przykładach jest wciąż wymagany. Jeśli natomiast wybrany schemat partycjonowania jest taki jak ten w Podręczniku, prefiks w niczym nie przeszkadza, należy tylko dodatkowo utworzyć odpowiednie dowiązanie symboliczne o nazwie boot. W skrócie: powyższe przykładowe konfiguracje będą działać bez względu na to czy utworzono osobną partycję /boot, czy nie.

Jeżeli przy uruchomieniu systemu zachodzi potrzeba przekazania do jądra dodatkowych opcji, wystarczy dopisać je na końcu polecenia kernel. W tym przykładzie został już dodany jeden parametr (root=/dev/hda3 lub real_root=/dev/hda3), który można wzbogacić o kolejne. Przykładowo posłużymy się opcją vga, dotyczącą bufora ramki, która została opisana wcześniej:

Użytkownicy jąder 2.6.7 i nowszych, którzy przy pomocy przełącznika ograniczyli wielkość dysku, ponieważ ich BIOS nie jest w stanie współpracować z większymi napędami mogą dodać opcję hdx=stroke, aby korzystać z całego dostępnego na urządzeniu miejsca.

Użytkownicy programu genkernel powinni wiedzieć, że jądra, które zbudowali używają tych samych opcji bootowania co LiveCD. Na przykład jeżeli komputer ma zainstalowane urządzenie SCSI, należy dodać parametr doscsi.

Następnie trzeba zapisać plik grub.conf i opuścić edytor. Kolejnym krokiem będzie dopisanie GRUB-a do MBR.

Twórcy GRUB-a zalecają użycie grub-install. Czasem jednak program ten z jakiegoś powodu odmawia współpracy. Mimo to, wciąż pozostaje możliwość ręcznego zainstalowania GRUB-a.

Teraz można przejść do paragrafu Domyślnie: Instalowanie GRUB-a przy pomocy grub-install lub Alternatywnie: Ręczne instalowanie GRUB-a.

Domyślnie: Instalacja GRUB-a przy pomocy grub-install

Aby zainstalować GRUB-a, należy wpisać komendę grub-install. Nie zadziała to jednak bez pewnych zmian, bo wciąż działamy w chrootowanym środowisku. Należy stworzyć plik /etc/mtab (plik z informacjami dotyczącymi zamontowanych napędów). Na szczęście jest na to prosta metoda, wystarczy skopiować plik /proc/mounts do /etc/mtab, pomijając jedynie linię rootfs - o ile nie stworzono osobnej partycji rozruchowej. W obu przypadkach zadziała następujące polecenie:

Listing 5: Tworzenie /etc/mtab

```
# grep -v rootfs /proc/mounts > /etc/mtab
```

Następnie przy pomocy grub-install zainstalowany zostaje GRUB:

Listing 6: Uruchamianie grub-install

```
# grub-install /dev/hda
```

Więcej informacji o GRUB-ie można znaleźć w dokumentach: GRUB FAQ i GRUB Manual.

Następnie należy przejść do części Ponowne uruchamianie systemu.

Alternatywnie: Ręczna instalacja GRUB-a

Aby zacząć, należy wpisać polecenie grub. Znak zachęty zmieni się na grub>. Teraz należy wprowadzić serię odpowiednich komend, aby GRUB został zapisany na dysku. Komendy te zostaną opisane poniżej.

Listing 7: Uruchamianie powłoki GRUB-a

```
# grub
```

Uwaga: Jeżeli komputer nie ma zainstalowanej stacji dyskietek, należy dodać opcję --no-floppy do powyższego polecenia, żeby GRUB nie marnował czasu na poszukiwanie nieistniejącego napędu.

Wyobraźmy sobie, że chcemy zainstalować GRUB tak aby odczytywał informacje z bootowej partycji /dev/hda1 i instalował swój rekord rozruchowy w MBR twardego dysku, tak żeby pierwszą rzeczą jaką zobaczymy po uruchomieniu komputera był znak zachęty GRUB-a. Oczywiście należy odpowiednio dostosować wszystkie opcje do używanej konfiguracji.

Mechanizm uzupełniania składni jest bardzo pomocny przy konfigurowaniu GRUB-a. Na przykład wpisując "root (" i wciskając TAB można zobaczyć wszystkie urządzenia (np. hd0). Jeśli zostanie wpisane "root (hd0," i potem zostanie naciśnięty TAB, ukaże się lista wszystkich dostępnych partycji na urządzeniu (np. hd0,0).

Przy pomocy tego systemu skonfigurowanie GRUB-a powinno być proste.

Listing 8: Instalacja GRUB-a w głównym sektorze rozruchowym

```
grub> root (hd0,0)          (Odpowiednia partycja /boot)
grub> setup (hd0)           (Instalacja GRUB-a w MBR)
grub> quit                  (Opuszczenie powłoki GRUB)
```

Uwaga: W przypadku konieczności zainstalowania GRUB poza MBR, należy zmienić komendę setup tak by wskazywała na odpowiednie urządzenie. Na przykład dla GRUB-a zainstalowanego w /dev/hda3 będzie to setup (hd0,2). Niewielu użytkowników decyduje się na skorzystanie z tej możliwości.

Więcej informacji można znaleźć na stronach: GRUB FAQ i GRUB Manual.

Następnie można przejść do sekcji Ponowne uruchamianie systemu.

10.c. Alternatywnie: Użycie LILO

LILO, czyli LInuxLOader, to program naprawdę wypróbowany i używany od bardzo dawna przez bardzo wielu użytkowników. Brakuje mu jednak niektórych cech posiadanych przez program GRUB, co jest, między innymi, powodem rosnącej popularności tego drugiego. Fakt, że na niektórych systemach LILO działa, a GRUB nie, powoduje, iż LILO jest ciągle w użyciu. Oczywiście nie jest to jedyny powód - niektórzy po prostu lepiej znają LILO i wolą przy nim pozostać, zamiast uczyć się obsługi całkiem nowego programu. My jak zawsze wybór pozostawiamy użytkownikom, Gentoo wspiera obydwa programy.

Instalacja LILO jest prosta, używamy do tego polecenia emerge.

Listing 9: Instalowanie LILO

```
# emerge lilo
```

Konfigurowanie LILO

Aby skonfigurować LILO, należy utworzyć plik /etc/lilo.conf. Można to zrobić poprzez uruchomienie ulubionego edytora (w tym Podręczniku konsekwentnie używamy nano) i w ten sposób stworzyć ten plik.

Listing 10: Tworzenie /etc/lilo.conf

```
# nano -w /etc/lilo.conf
```

Kilka rozdziałów temu prosiliśmy o zapamiętanie nazwy utworzonego obrazu jądra. W naszym przykładowym lilo.conf użyty zostanie uprzednio zaprezentowany schemat partycjonowania. Przygotowane zostały dwie części:

- Pierwsza dla tych, którzy nie skorzystali z programu genkernel przy budowaniu jądra
- Druga dla używających genkernela.

Należy znać nazwy swojego obrazu jądra i swojego obrazu initrd.

Uwaga: If system plików głównej partycji to JFS, należy dodać "ro" do linii kernel w związku z tym, że JFS musi odtworzyć swój log zanim pozwoli na montowanie w trybie do odczytu i zapisu.

Listing 11: Example /etc/lilo.conf

```
boot=/dev/hda          # Instalacja LILO w MBR
prompt                 # Dajemy użytkownikowi możliwość wyboru innej pozycji
timeout=50              # Czekamy 5 sekund przed uruchomieniem domyślnej pozycji
default=gentoo          # Kiedy oczekiwanie się zakończy, uruchamiana jest pozycja
"gentoo"

# Dla osób nie korzystających z genkernela
image=/boot/kernel-2.6.17-gentoo-r5
label=gentoo            # Nazwa tej części
read-only               # Tryb tylko do odczytu na starcie, nie zmieniać!
root=/dev/hda3          # Miejsce, w którym znajduje się partycja główna

# Dla użytkowników genkernela
image=/boot/kernel-genkernel-x86-2.6.17-gentoo-r5
label=gentoo
read-only
root=/dev/ram0
append="init=/linuxrc ramdisk=8192 real_root=/dev/hda3 udev"
initrd=/boot/initramfs-genkernel-x86-2.6.17-gentoo-r5

# Następne dwie linie są podane na wypadek, gdybyśmy chcieli uruchamiać także system
Windows.
# W tym przypadku, Windows mieści się na /dev/hda6.
other=/dev/hda6
label=windows
```

Uwaga: Parametr udev na końcu linii "append" jest niezbędny dla obejścia błędu w pewnych wersjach genkernela, występującego gdy używa się udev na pierwszym miejscu, co jest zresztą domyślnym ustawieniem.

Uwaga: Jeśli używany jest inny schemat partycjonowania i/lub obrazu jądra, należy dokonać niezbędnych poprawek.

Jakiekolwiek dodatkowe parametry startowe jądra dodaje się po słowie kluczowym append. Jako przykład, podajemy opcję video służącą do włączenia bufora ramki:

Listing 12: Dodawanie funkcji jądra poprzez linię append

```
image=/boot/kernel-2.6.17-gentoo-r5
label=gentoo
read-only
root=/dev/hda3
append="video=vesafb:mtrr,ywrap,1024x768-32@85"
```

Dla jądra 2.6.7 lub nowszego i ograniczenia przy pomocy przełączników rozmiaru twardego dysku, gdy BIOS nie obsługuje tak dużych napędów, należy dopisać do linii append polecenie hdx=stroke.

Użytkownicy programu genkernel powinni wiedzieć, że jądra, które zbudowali, używają tych samych opcji bootowania co płyty instalacyjne. Na przykład w przypadku korzystania z urządzenia SCSI należy dodać parametr dosesi.

Po zakończonej konfiguracji, należy zapisać plik i opuścić edytor. By zakończyć wystarczy uruchomić program /sbin/lilo, LILO zastosuje ustawienia z /etc/lilo.conf (tzn. zainstaluje się na dysku). Przy każdej zmianie w /etc/lilo.conf oraz przy zmianie jądra, trzeba ponownie wykonać polecenie /sbin/lilo.

Listing 13: Kończenie instalacji LILO

```
# /sbin/lilo
```

10.d. Ponowne uruchamianie systemu

Należy opuścić chroot i odmontować wszystkie zamontowane partycje. Następnie trzeba wpisać komendę reboot.

Listing 14: Odmontowywanie partycji i ponowne uruchamianie komputera

```
# exit  
cdimage ~# cd  
cdimage ~# umount /mnt/gentoo/boot /mnt/gentoo/dev /mnt/gentoo/proc /mnt/gentoo  
cdimage ~# reboot
```

Trzeba pamiętać o usunięciu płyty z napędu, bo zamiast nowutkiego Gentoo znów zobaczymy system z płyty instalacyjnej.

Następnie wystarczy zakończyć proces instalacji zgodnie ze wskazówkami opisanymi w rozdziale [Finalizowanie instalacji Gentoo](#).

11. Zakończenie instalacji Gentoo

11.a. Administrowanie kontami użytkowników

Tworzenie konta do codziennej pracy

Wykonywanie zadań z przywilejami roota jest niebezpieczne i należy tego unikać. Do codziennej pracy należy utworzyć zwykłe konto użytkownika.

Czynności jakie może wykonać użytkownik są zależne od grup do jakich należy. Oto lista najważniejszych grup:

Grupa	Opis
audio	Dostęp do urządzeń audio
cdrom	Bezpośredni dostęp do urządzeń optycznych
floppy	Bezpośredni dostęp do stacji dyskietek
games	Możliwość uruchomienia gier
portage	Daje możliwość korzystania z polecenie emerge --pretend przez zwykłego użytkownika.
usb	Dostęp do urządzeń USB
plugdev	Umożliwia montowanie i używanie przenośnych urządzeń takich jak pamięci podręczne USB czy aparaty fotograficzne.
video	Możliwość dostępu do urządzeń video oraz pracy z akceleracją sprzętową
wheel	możliwość używania polecenia su

Na przykład, aby utworzyć konto użytkownika mkay i dodać go do grup wheel (możliwość korzystania z su do przełączania się na konto root), users (grupa domyślana dla wszystkich użytkowników) oraz audio (możliwość korzystania z urządzeń dźwiękowych) należy z konta roota wykonać następujące polecenie:

Listing 1: Dodawanie użytkownika do codziennej pracy

```
Login: root
Password: (wpisujemy hasło)

# useradd mkay -m -G users,wheel,audio -s /bin/bash
# passwd mkay
Password: (hasło mkaya)
Re-enter password: (Ponownie hasło mkaya)
```

Jeśli użytkownik ten kiedykolwiek zechce wykonać jakiekolwiek czynności jako root powinien użyć polecenia su -, aby tymczasowo otrzymać uprawnienia superużytkownika. Alternatywnie może skorzystać z pakietu sudo charakteryzującego się wysokim poziomem bezpieczeństwa (o ile zostanie prawidłowo skonfigurowany).

12. I co dalej?

12.a. Dokumentacja

Gratulacje! Mamy już działający system Gentoo Linux. Ale... co teraz? Czego można dokonać? Co odkryć najpierw? Gentoo daje swoim użytkownikom ogromne możliwości, których większość jest świetnie udokumentowana.

Zdecydowanie warto rzucić okiem na drugą część Podręcznika, zatytułowaną Praca z Gentoo. Omówione w niej zostały metody instalacji i aktualizacji oprogramowania, flagi USE i system skryptów startowych.

Aby zoptymalizować system na desktop lub dowiedzieć się jak najlepiej skonfigurować oprogramowanie biurkowe, warto poznać rozdział Zasoby dokumentacji Gentoo dla stacji roboczych. Warto również zainteresować się możliwością spolszczenia systemu. Wszystkie czynności, jakich należy dokonać w tym celu opisaliśmy w tekście zatytułowanym Lokalizacja Gentoo Linux.

Wartą przeczytania pozycją jest także Podręcznik bezpieczeństwa Gentoo.

Pełna lista dostępnych dokumentów znajduje się na stronie zasobów dokumentacji Gentoo.

12.b. Gentoo w sieci

Wszystkich użytkowników zapraszamy na [Forum Gentoo](#) oraz nasze liczne kanały IRC.

Dodatkowo posiadamy wiele [list dyskusyjnych](#). Informacje o subskrypcji zamieściliśmy na ich stronie.

B. Praca z Gentoo

1. Wprowadzenie do Portage

1.a. Witamy w Portage

Portage to najlepszy istniejący program do zarządzania oprogramowaniem. Żadna inna dystrybucja Linuksa nie może się pochwalić równie kompleksowym, konfigurowalnym i użytecznym narzędziem jak to napisane przez deweloperów Gentoo.

Portage zostało napisane w dwóch językach skryptowych, Pythonie i Bashu, dzięki czemu sposób jego działania jest bardzo przejrzysty nawet dla niezbyt biegłych w programowaniu użytkowników.

Większość użytkowników pracuje z Portage przy pomocy narzędzia emerge. Aby uzyskać więcej informacji na temat tego programu, wystarczy wpisać:

Listing 1: Czytanie man emerge

```
$ man emerge
```

1.b. Drzewo Portage

Ebuildy

Kiedy mówimy o pakietach to tak naprawdę mamy na myśli programy dostępne dla użytkowników Gentoo w drzewie Portage. Drzewo to jest zbiorem ebuildów, czyli plików zawierających wszelkie informacje, które są niezbędne do zarządzania oprogramowaniem (instalacja, wyszukiwanie, inne zapytania...). Domyślnie kolekcja ebuildów znajduje się w katalogu /usr/portage.

Za każdym razem gdy zażądamy od Portage wykonania jakiegoś zadania związanego z naszym oprogramowaniem użyje ono jako podstawy swojego działania informacji zawartych w kolekcji ebuildów. Stąd też warto w miarę często uaktualniać swoje drzewo Portage tak, aby system wiedział o nowych wersjach programów, poprawkach do nich, etc.

Uaktualnianie drzewa Portage

Drzewo Portage uaktualniamy zazwyczaj za pomocą narzędzia [rsync](#). Uaktualnienie to wykonuje się w stosunkowo prosty sposób dzięki jednemu z parametrów polecenia emerge, dzięki któremu komenda ta zadziała jak nakładka na rsync:

Listing 2: Uaktualnianie drzewa Portage

```
# emerge --sync
```

Jeśli nie jest możliwe użycie rsync w wyniku jakichś ograniczeń narzuconych przez różnego rodzaju firewalle to możliwa jest aktualizacja drzewa Portage przy użyciu jednego z generowanych codziennie snapshotów. Program emerge-webrsync automatycznie pobierze odpowiednie pliki i zainstaluje je w systemie.

Listing 3: Uruchamianie emerge-webrsync

```
# emerge-webrsync
```

1.c. Zarządzanie oprogramowaniem

Wyszukiwanie oprogramowania

Do wyszukiwania w drzewie Portage konkretnych programów można użyć funkcji wbudowanych w program emerge. Domyślnie emerge --search wypisze wszystkie zawierające dane wyrażenie nazwy pakietów.

Na przykład poszukajmy wszystkich pakietów zawierających literki "pdf" w nazwie:

Listing 4: Wyszukiwanie pakietów z pdf w nazwie

```
$ emerge --search pdf
```

By przeszukiwać pakiety również po opisie pakietu, nie tylko po jego nazwie należy dopisać dodatkowo parametr --searchdesc (lub krócej -S).

Listing 5: Wyszukiwanie wszystkich związanych z pdf paczek

```
$ emerge --searchdesc pdf
```

Kiedy przyjrzymy się wynikowi tego polecenia zauważymy, że dostarcza on wielu ciekawych informacji. Zawartość i opisy poszczególnych pól są dość przejrzyste i nie powinny przysporzyć nikomu problemów. Z tego względu nie będziemy ich tu szerzej omawiać.

Listing 6: Przykładowy wynik polecenia emerge --search

```
* net-print/cups-pdf
    Latest version available: 1.5.2
    Latest version installed: [ Not Installed ]
    Size of downloaded files: 15 kB
    Homepage: http://cip.physik.uni-wuerzburg.de/~vrbehr/cups-pdf/
    Description: Provides a virtual printer for CUPS to produce PDF files.
    License: GPL-2
```

Instalowanie oprogramowania

Instalacja znalezionych w ten sposób w Portage programów jest prosta i sprowadza się do dodania do polecenia emerge nazwy programu do zainstalowania. Dla przykładu zainstalujemy sobie gnumeric:

Listing 7: Instalacja gnumeric

```
# emerge gnumeric
```

W związku z tym, że wiele aplikacji do prawidłowego działania potrzebuje innych programów, instalacja którejś paczki może nieść ze sobą potrzebę zainstalowania także jej zależności. Nie ma powodu do zmartwień, to nie RPM-y - Portage doskonale radzi sobie z zależnościami. By dowiedzieć się, jakie zależności zostaną zainstalowane z danym programem należy dodać przełącznik --pretend do zwykłej komendy instalującej program. Na przykład:

Listing 8: Udajemy, że chcemy zainstalować gnumeric

```
# emerge --pretend gnumeric
```

Kiedy zostanie wydane polecenie dla Portage by zainstalowało jakiś program, z Internetu zostaną pobrane wszystkie niezbędne, nieznajdujące się na dysku pliki zawierające kod źródłowy. Domyślnie są one przechowywane w katalogu /usr/portage/distfiles. Następnie program zostanie rozpakowany, skompilowany i zainstalowany. Aby Portage jedynie pobrało potrzebne pliki, należy dodać opcję --fetchonly do komendy emerge:

Listing 9: Pobieranie kodu źródłowego gnumeric

```
# emerge --fetchonly gnumeric
```

Wyszukiwanie dokumentacji do zainstalowanych pakietów

Wiele pakietów jest publikowanych jest wraz z dokumentacją. Czasem flaga USE doc określa, czy dokumentacja dla danego pakietu zostanie zainstalowana czy nie. Informację o tym, czy dany pakiet korzysta z flagi doc można uzyskać za pomocą następującego polecenia: emerge -vp <nazwa pakietu>.

Listing 10: Sprawdzenie czy pakiet używa flagi doc.

```
(Oczywiście alsa-lib to tylko przykład)
# emerge -vp alsa-lib
[ebuild N ] media-libs/alsa-lib-1.0.9_rc3 +doc -jack 674 kB
```

Flagę doc można włączyć bądź wyłączyć globalnie w pliku /etc/make.conf lub dla poszczególnych pakietów w pliku /etc/portage/package.use. Możemy również stworzyć katalog /etc/portage/package.use gdzie umieszczamy pliki z flagami USE. Rozdział Flagi USE omawia ten temat szczegółowo.

Dokumentacja do zainstalowanego już pakietu na ogół znajduje się w podkatalogu o nazwie takiej samej jak pakiet, w katalogu /usr/share/doc. Można wyświetlić listę wszystkich zainstalowanych plików za pomocą narzędzia equery, które jest częścią pakietu app-portage/gentoolkit.

Listing 11: Lokalizowanie dokumentacji pakietu

```
# ls -l /usr/share/doc/alsa-lib-1.0.9_rc3
total 28
-rw-r--r-- 1 root root 669 May 17 21:54 ChangeLog.gz
-rw-r--r-- 1 root root 9373 May 17 21:54 COPYING.gz
drwxr-xr-x 2 root root 8560 May 17 21:54 html
-rw-r--r-- 1 root root 196 May 17 21:54 TODO.gz

(Można też użyć equery do zlokalizowania plików dokumentacji)
# equery files alsa-lib | less
media-libs/alsa-lib-1.0.9_rc3
* Contents of media-libs/alsa-lib-1.0.9_rc3:
/usr
/usr/bin
/usr/bin/alsalisp
(Wyjście programu zostało skrócone)
```

Usuwanie oprogramowania

Do usuwania zainstalowanych programów służy polecenie emerge --unmerge. Nakaże ono Portage usunięcie wszystkich plików dodanych w procesie instalacji programu, z pominięciem jednak tych plików, które od instalacji programu zostały zmienione. Najczęściej chodzi tu o pliki konfiguracyjne, a pozostawienie ich na dysku umożliwia łatwe wznowienie pracy z programem w przypadku, gdy w przyszłości program zostanie ponownie zainstalowany.

W tym dość przejrzystym procesie kryje się pewna pułapka: Portage nie sprawdza, czy pakiet, który ma być usunięty nie jest zależnością innego zainstalowanego programu. Jeśli jednak jest to program niezbędny dla prawidłowego działania systemu, pojawi się ostrzeżenie.

Listing 12: Usuwanie gnumeric z systemu

```
# emerge --unmerge gnumeric
```

Gdy program zostanie usunięty, jego zależności nie są usuwane razem z nim, ale pozostają na dysku. Aby odszukać i usunąć niepotrzebne w systemie zależności używamy polecenia emerge --depclean. Omówimy je dokładniej nieco później.

Uaktualnianie systemu

Aby utrzymać swój system w dobrej kondycji (nie wspominając już o instalacji najnowszych poprawek związanych z bezpieczeństwem), należy dość często go uaktualniać. W związku z tym, że w tym procesie Portage porównuje zainstalowane oprogramowanie z ebuildami z drzewa Portage, należy najpierw pobrać jego aktualną wersję. Kiedy już je zaktualizujemy przychodzi czas na właściwe uaktualnienie systemu. Dokonujemy tego poleceniem emerge --update world. W poniższym przykładzie skorzystamy także z opcji --ask, która spowoduje wyświetlenie listy pakietów do aktualizacji, a następnie pytania czy na pewno chcemy je zaktualizować.

Listing 13: Uaktualnianie systemu

```
# emerge --update --ask world
```

Portage znajdzie wszystkie bezpośrednio zainstalowane przez użytkownika aplikacje, dla których są dostępne nowsze wersje, ale pomimo aktualnienia ich zależności. Aby uaktualnić całe oprogramowanie wraz z zależnościami, należy dodać jeszcze argument --deep:

Listing 14: Uaktualnienie całego systemu

```
# emerge --update --deep world
```

W związku z tym, że poprawki związane z bezpieczeństwem zdarzają się nie tylko w programach zainstalowanych bezpośrednio, ale również w ich zależnościach zalecamy częste uruchamianie tego polecenia.

Jeżeli ostatnio zmieniane były flagi USE, polecamy również dodanie do całej tej linii polecień argumentu --newuse. Portage sprawdzi wtedy czy zmiany we flagach USE niosą ze sobą potrzebę przekompilowania i przeinstalowania których z zainstalowanych programów:

Listing 15: Przeprowadzenie pełnego uaktualnienia

```
# emerge --update --deep --newuse world
```

Metapakiety

Niektóre z pakietów w drzewie Portage nie mają żadnej zawartości, ale służą do instalacji całych kolekcji innych pakietów. Doskonałym przykładem takiego zestawu jest pakiet KDE, który służy do instalowania kompletnego środowiska graficznego. Możemy dzięki jego istnieniu przy pomocy jednego polecenia dodać do systemu wszystkie programy, biblioteki oraz zależności związane z KDE.

Jeśli kiedykolwiek zdarzy nam się posiadać taki pakiet zainstalowany w systemie, będziemy mieli pewien problem z jego odinstalowaniem. Zwykle wpisanie emerge --unmerge poczyni stosunkowo małe spustoszenie w niepotrzebnych nam już plikach, ponieważ ogromna ilość zależności pozostanie w systemie.

Portage jest w stanie poradzić sobie z tego typu "osieroconymi" zależnościami, ale najpierw należy w pełni uaktualnić swój system, uwzględniając przy tym również zmiany we flagach USE. Następnie uruchamiamy wspomniane już wcześniej polecenie emerge --depclean, aby usunąć "osierocone" zależności, a kiedy już skończymy je odinstalowywać przebudowujemy wszystkie programy, które wcześniej były dynamicznie z nimi zlinkowane, a teraz już ich nie potrzebują.

Cały proces sprawdza się do wpisania trzech prostych polecień:

Listing 16: Usuwanie osieroconych zależności

```
# emerge --update --deep --newuse world
# emerge --depclean
# revdep-rebuild
```

Program revdep-rebuild znajduje się w pakiecie gentoolkit wraz z kilkoma innymi bardzo przydatnymi programami. Aby używać programu, należy oczywiście najpierw zainstalować ten pakiet.

Listing 17: Instalacja pakietu gentoolkit

```
# emerge gentoolkit
```

1.d. Kiedy Portage narzeka

...na sloty, virtuale, gałęzie, architektury i profile

Jak już wcześniej zaznaczaliśmy, Portage jest potężnym narzędziem i posiada możliwości jakich nie ma żaden inny program do zarządzania oprogramowaniem. Postaramy się teraz w skrócie przedstawić kilka aspektów pracy z Portage.

W Portage możliwe jest posiadanie kilku różnych wersji jednego programu. Podczas gdy inne dystrybucje obchodzą problem nadając po prostu takim pakietom różne numery porządkowe, jak np. freetype i freetype2 Portage wykorzystuje tzw. slotów.

Każdy ebuild posiada osobny slot dla wersji programu, którą reprezentuje, więc ebuildy różnych wersji programu mogą koegzystować w jednym systemie. Na przykład paczka freetype posiada ebuildy z ustawionymi wartościami SLOT="1" i SLOT="2".

Są również pakietы, które wykonują te same czynności, ale w różny sposób. Doskonałym przykładem takiego programu są loggery systemowe: metalogd, sysklogd i syslog-ng. Aplikacje, które do prawidłowego działania potrzebują loggera systemowego nie mogą posiadać w zależnościach jedyne np. metalogd, ponieważ pozostałe programy z tej grupy również są w stanie spełnić tę zależność. Do tego właśnie służą Virtuale. Każdy z loggerów systemowych dostarcza po prostu virtual/syslog, który jest jednocześnie zależnością dla innych programów.

Oprogramowanie znajdujące się w drzewie Portage jest podzielone na gałęzie. Domyślnie używana jest gałąź stabilna dla danej architektury. Nowe i nieprzetestowane programy są dodawane do gałęzi niestabilnej, czyli testowej. Dopóki ich niezawodność nie zostanie potwierdzona i nie zostaną przeniesione do gałęzi stabilnej, Portage nie zainstaluje ich, chociaż ebuildy nowszych wersji będą się znajdowały w drzewie.

Niektóre programy są dostępne tylko dla określonych architektur. Czasem na innych wcale nie działają, czasem potrzebują jeszcze nieco testów, może się też zdarzyć, że deweloper danego programu nie ma po prostu czasu lub możliwości, aby przetestować taki pakiet na różnych architekturach.

Każdej instalacji Gentoo przypisany jest określony profil, który zawiera między innymi listę pakietów, które są niezbędne do prawidłowego działania systemu.

Zablokowane pakietы

Listing 18: Ostrzeżenie przed blokadą pakietu w Portage (z opcją --pretend)

```
[blocks B      ] mail-mta/ssmtp (is blocking mail-mta/postfix-2.2.2-r1)
```

Listing 19: Ostrzeżenie Portage przed blokadą pakietu (bez opcji --pretend)

```
!!! Error: the mail-mta/postfix package conflicts with another package.
!!!          both can't be installed on the same system together.
!!!          Please use 'emerge --pretend' to determine blockers.
```

W ebuildach znajdują się określone pola, które informują Portage na temat zależności danego programu. Są dwa rodzaje takich zależności: Zależności niezbędne do zbudowania programu - deklarowane przez DEPEND oraz zależności niezbędne do jego uruchomienia - deklarowane jako RDEPEND. Kiedy któraś z tych zależności jest niekompatybilna z jakimś virtuałem lub pakietem, jest włączana blokada.

Są dwie możliwości na pozbycie się blokady: Nie instalować programu lub usunąć pakiet, który go blokuje. W podanym powyżej przykładzie mogliśmy wybrać pomiędzy rezygnacją z instalacji postfix lub usunięciem ssmtp.

Może również się zdarzyć, że blokują się pakietы, które nie są jeszcze zainstalowane. W takim rzadkim przypadku należy się dokładnie zastanowić czemu oba mają być zainstalowane. Zwykle można sobie poradzić tylko z jednym z tych pakietów. Jeśli nie jest to możliwe prosimy o zgłoszenie błędu.

Zamaskowane pakietы

Listing 20: Ostrzeżenie Portage o zamaskowanych pakietach

```
!!! all ebuilds that could satisfy "bootsplash" have been masked.
```

Listing 21: Ostrzeżenie Portage o zamaskowanych pakietach - z podaniem przyczyny

```
!!! possible candidates are:
```

- gnome-base/gnome-2.8.0_pre1 (masked by: ~x86 keyword)
- lm-sensors/lm-sensors-2.8.7 (masked by: -sparc keyword)
- sys-libs/glibc-2.3.4.20040808 (masked by: -* keyword)
- dev-util/cvsd-1.0.2 (masked by: missing keyword)
- media-video/ati-gatos-4.3.0 (masked by: package.mask)
- sys-libs/glibc-2.3.2-r11 (masked by: profile)

Jeśli zechcemy zainstalować paczkę, która nie jest dostępna dla naszego systemu dostaniemy właśnie taki komunikat. Możemy wtedy zainstalować inny spełniający te same funkcje, ale dostępny dla naszego systemu program lub poczekać aż pakiet zostanie

odmaskowany.

Maskowanie pakietów nie odbywa się bez przyczyny:

- Słowo kluczowe **~arch** oznacza, że aplikacja nie została jeszcze dostatecznie sprawdzona na naszej architekturze, aby znaleźć się w gałęzi stabilnej. Zwykle w takim przypadku wystarczy poczekać kilka dni (rzadziej tygodni) i spróbować ponownej jej instalacji.
- Słowo kluczowe **-arch** lub **-*** oznacza, że program nie działa na naszej architekturze. Jeśli jednak aplikacja działa i są dowody na poparcie tej tezy prosimy o zgłoszenie tego na naszą [Bugzille](#).
- Komunikat **missing keyword** oznacza, że aplikacja nie została jeszcze przetestowana na tej architekturze. W takim przypadku należy poprosić któregoś z developerów zajmujących się tymi sprawami o przetestowanie pakietu lub uczynić to własnoręcznie i zgłosić wyniki swoich badań na [Bugzille](#).
- Komunikat **package.mask** oznacza, że pakiet jest uszkodzony, niestabilny lub co gorsza w ogóle nie nadaje się do użytku.
- Komunikat z tekstem **profile** oznacza, że pakiet nie pasuje do naszego profilu systemowego i gdybyśmy go zainstalowali mógłby zepsuć nasz system.

Brakujące zależności

Listing 22: Komunikat Portage o brakujących zależnościach

```
emerge: there are no ebuilds to satisfy ">=sys-devel/gcc-3.4.2-r4".
!!! Problem with ebuild sys-devel/gcc-3.4.2-r2
!!! Possibly a DEPEND/*DEPEND problem.
```

Aplikacja, którą próbujemy zainstalować jest zależna od pakietu, który nie jest dostępny dla danej architektury. Należy sprawdzić na [Bugzilli](#) czy problem został już zgłoszony i ewentualnie go zgłosić, jeśli nie zrobił tego ktoś inny. Jeśli nie są mieszane różne typy gałęzi Portage w jednym systemie to problem ten nie powinien wystąpić i zwykle oznacza błąd w drzewie.

Niejasna nazwa pakietu

Listing 23: Ostrzeżenie Portage dotyczące niejasnych nazw pakietów

```
!!! The short ebuild name "aterm" is ambiguous. Please specify
!!! one of the following fully-qualified ebuild names instead:
    dev-libs/aterm
    x11-terms/aterm
```

Program, który próbujemy zainstalować ma nazwę, którą posiada więcej niż jeden pakiet. Aby rozwiązać ten problem wystarczy dokładniej sprecyzować co chcemy zainstalować dodając przed nazwą programu kategorię, do której on należy.

Wzajemnie od siebie zależne pakiety

Listing 24: Ostrzeżenie Portage na temat wzajemnie od siebie zależnych pakietów

```
!!! Error: circular dependencies:
ebuild / net-print/cups-1.1.15-r2 depends on ebuild / app-text/ghostscript-7.05.3-r1
ebuild / app-text/ghostscript-7.05.3-r1 depends on ebuild / net-print/cups-1.1.15-r2
```

Sprawa jest prosta. Dwa pakiety (lub więcej), które próbujemy zainstalować są od siebie wzajemnie zależne i w związku z tym nie mogą zostać zainstalowane. Oznacza to błąd w drzewie Portage, który zostanie usunięty możliwie najszybciej od momentu jak pierwszy użytkownik zgłosi ten problem na Bugzille.

Niedane pobieranie

Listing 25: Komunikat Portage o nieudanym pobieraniu

```
!!! Fetch failed for sys-libs/ncurses-5.4-r5, continuing...
(...)
!!! Some fetch errors were encountered. Please see above for details.
```

Oznacza to, że Portage nie było w stanie pobrać źródeł żądanej aplikacji, w związku z czym zostało zmuszone do zrezygnowania z jej instalacji i będzie instalowało kolejne programy z listy. Błąd najczęściej jest spowodowany wstawieniem złego adresu serwera w ebuildzie programu lub dlatego, że serwer lustrzany nie zdążył jeszcze się zsynchronizować. Możliwa jest również sytuacja, że serwer, na którym znajdują się źródła, z jakichś względów jest nieczynny.

Należy odczekać około godziny i spróbować ponownie zainstalować program.

Ochrona profilu systemu

Listing 26: Ostrzeżenie Portage dotyczące pakietu chronionego profilem systemowym

```
!!! Trying to unmerge package(s) in system profile. 'sys-apps/portage'  
!!! This could be damaging to your system.
```

Taki komunikat oznacza, że pakiet, który próbujemy usunąć jest kluczowy dla działania systemu. Znajduje się on na liście profilu systemowego jako niezbędny i w związku z tym nie zostanie usunięty.

2. Flagi USE

2.a. Czym są flagi USE?

Idea flag USE

Kiedy instalujemy Gentoo (lub dowolną inną dystrybucję, albo nawet inny system operacyjny) zwykle dokonujemy wyborów zależnych od środowiska, w którym przychodzi nam pracować. Instalacja dla serwera różni się od instalacji dla stacji roboczej. Konfiguracja komputera dla gracza różni się od tej dla komputera przeznaczonego do obróbki grafiki 3D.

Nie jest tak tylko w przypadku pakietów, które wybieramy przy instalacji, ale także dla cech, które dany pakiet powinien posiadać. Jeżeli nie potrzebujemy obsługi OpenGL, dlaczego mielibyśmy instalować OpenGL oraz jego obsługę w większości pakietów? Jeżeli nie chcemy używać KDE, dlaczego mamy budować pakiety ze wsparciem dla KDE, podczas gdy bez problemów może pracować bez niego?

Aby ułatwić użytkownikom decydowanie o tym czego potrzebują, a czego nie chcą instalować i aktywować stworzyliśmy dla nich specjalne środowisko. Dzięki niemu użytkownik może wybrać to co jest mu potrzebne, a Portage znacznie ułatwi mu cały proces wybierania najlepszych ustawień.

Definicja flag USE

Każda flaga jest słowem kluczowym, które reprezentuje wspierane funkcje oraz informacje o zależnościach dla wybranego wątku. Jeżeli zdefiniujemy jakąś flagę USE Portage będzie wiedziało, że jest nam potrzebne wsparcie funkcji przypisanej temu słowi kluczowemu. Oczywiście uwzględnione zostaną także pakiety zależne.

Przyjrzyjmy się zatem przykładowi: słowi kluczowemu kde. Jeżeli nie posiadamy go wśród zmiennych USE wszystkie pakiety, które posiadają opcjonalną obsługę KDE zostaną skompilowane bez obsługi KDE. Wszystkie pakiety, które będą opcjonalnie zależne od KDE, zostaną zainstalowane bez bibliotek KDE jako zależności. Jeżeli zdefiniujemy słowo kluczowe kde, to te pakiety zostaną skompilowane z obsługą KDE oraz biblioteki KDE zostaną zainstalowane jako pakiety zależne.

Dzięki dobremu doborowi słów kluczowych otrzymamy system dokładnie dostosowany do naszych potrzeb.

Jakie wyróżniamy flagi USE?

Wyróżniamy dwa typy flag USE: globalne oraz lokalne.

- Globalne flagi USE są używane dla większej ilości pakietów, są ogólnosystemowe. Większość ludzi postrzega je właśnie jako flagi USE.
- Lokalne flagi USE są używane przez pojedynczy pakiet w celu podjęcia decyzji specyficznych dla danego pakietu.

Lista dostępnych globalnych flag USE jest dostępna w Internecie lub też lokalnie w pliku /usr/portage/profiles/use.desc.

Lista dostępnych lokalnych flag USE znajduje się w pliku /usr/portage/profiles/use.local.desc.

2.b. Używanie flag USE

Deklarowanie stałych flag USE

Kiedy już odkryliśmy jak ważny jest właściwy dobór flag USE możemy przystąpić do omawiania tego jak się je deklaruje.

Jak już wcześniej wspominaliśmy, wszystkie flagi USE są deklarowane wewnątrz zmiennej USE. Aby ułatwić użytkownikom szukanie oraz wybór flag USE dostarczamy dobrany przez nas domyślny zestaw. Zestaw ten jest kolekcją flag, które według nas są najczęściej wybierane przez użytkowników Gentoo. Domyślny zestaw jest zadeklarowany w pliku make.defaults i jest częścią wybranego profilu.

Profil, którego system używa jest wskazywany przez dowiązanie symboliczne /etc/make.profile. Każdy profil działa ponad innym, większym profilem, końcowy wynik jest więc sumą wszystkich profili. Górnny profil to base (/usr/portage/profiles/base).

Rzućmy okiem na te domyślne ustawienia dla profilu 2004.3:

Listing 1: Skumulowana zmienna USE dla profilu 2004.3

```
(Ten przykład to suma ustawień w plikach base, default-linux,
default-linux/x86 i default-linux/x86/2004.3)
USE="x86 oss arts avi berkdb bitmap-fonts crypt cups encode fortran f77
fomaticdb gdbm gif gpm gtk imlib jpeg kde gnome libg++ libwww mad
mikmod motif mpeg ncurses nls oggvorbis opengl pam pdfplib png python qt
quicktime readline sdl spell ssl svga tcpd truetype X xml2 xmms xv zlib"
```

Jak łatwo zauważyc domyślny zestaw zawiera dość dużo słów kluczowych. Pamiętajmy, aby **nie** dokonywać zmian w pliku make.defaults, w celu dostosowywania zmiennej USE do swoich potrzeb. Zmiany te zostaną usunięte przy najbliższej aktualizacji Portage!

Aby zmienić domyślne ustawienia, musimy dodać (lub usunąć) słowa kluczowe w zmiennej USE. Dokonuje się tego definiując globalnie zmienną USE w pliku /etc/make.conf. Do tej zmiennej możemy dodać flagi, które są nam potrzebne lub też usunąć te, których nie potrzebujemy. Usunięcia flagi dokonuje się poprzez wstawienie znaku minus (-) przed wybraną flagą.

Na przykład, aby usunąć obsługę KDE i QT oraz dodać obsługę ldap, zmienna USE w pliku /etc/make.conf powinna wyglądać następująco:

Listing 2: Przykładowe ustawienia zmiennej USE w pliku /etc/make.conf

```
USE="-kde -qt ldap"
```

Deklarowanie flag USE tylko dla wybranego pakietu

Czasami mamy zamiar zadeklarować wybraną flagę USE dla jednej (czasem kilku) aplikacji, ale nie dla całego systemu. Aby tego dokonać, będziemy zmuszeni do utworzenia katalogu /etc/portage (jeżeli nie istnieje) i wyedytowania pliku /etc/portage/package.use.

Na przykład, jeżeli nie chcemy globalnego wsparcia dla berkdb, ale chcielibyśmy mieć jego wsparcie dla mysql, powinniśmy dodać:

Listing 3: Przykład /etc/portage/package.use

```
dev-db/mysql berkdb
```

Oczywiście możemy całkowicie wyłączyć flagi USE dla wybranej aplikacji. Na przykład, jeżeli nie chcemy obsługi javy w PHP:

Listing 4: 2 przykład /etc/portage/package.use

```
dev-php/php -java
```

Deklarowanie tymczasowych flag USE

Czasami zachodzi potrzeba użycia flagi USE tylko jeden raz. Zamiast dwukrotnego edytowania pliku /etc/make.conf (aby wprowadzić, a potem cofnąć zmiany w USE) możemy po prostu zadeklarować tą flagę jako zmienną środowiskową. Pamiętajmy jednak, że jeżeli ponownie zainstalujemy lub zaktualizujemy daną aplikację (przypadkowo lub przy aktualizacji systemu) to takie zmiany nie zostaną ponownie wprowadzone.

Dla przykładu usuniemy obsługę javy na czas instalacji mozilli.

Listing 5: Używanie USE jako zmiennej środowiskowej

```
# USE="-java" emerge mozilla
```

Pierwszeństwo

Oczywiście istnieje pierwszeństwo w przydzielaniu priorytetów konkretnym flagom USE. Nie ma sensu deklarować zmiennej USE="-java" tylko po to, aby zobaczyć, że java i tak zostanie użyta w związku z zadeklarowaniem na wyższym poziomie. Hierarchia flag USE prezentuje się następująco (pierwsze pozycje mają najwyższy priorytet):

1. Domyślne ustawienia zmiennej USE znajdujące się w pliku make.defaults będącym częścią wybranego profilu
2. Zdefiniowana przez użytkownika zmienna USE znajdująca się w pliku /etc/make.conf
3. Zdefiniowana przez użytkownika zmienna USE w pliku /etc/portage/package.use
4. Zmienna USE zdefiniowana przez użytkownika jako zmienna środowiskowa.

Aby sprawdzić ostateczne ustawienia zmiennej USE widziane przez Portage wpisujemy polecenie emerge --info. Polecenie to wskaże wszystkie istotne zmienne (włączając zmienną USE) z wartościami używanymi aktualnie przez Portage.

Listing 6: Wykonywanie polecenia emerge --info

```
# emerge --info
```

Adaptacja systemu do nowych flag USE

Jeżeli zmodyfikowaliśmy flagi USE i chcemy uaktualnić system tak, aby pakiety używały nowych flag USE musimy uruchomić emerge z opcją --newuse.

Listing 7: Rekomplikacja systemu

```
# emerge --update --deep --newuse world
```

Następnie uruchamiamy depclean, który usunie niepotrzebne zależności, które zostały zemergowane na "starym" systemie, ale są nieaktualne z nowymi flagami USE.

Ostrzeżenie: Uruchomienie emerge --depclean jest niebezpieczną operacją i powinno być wykonywane z zachowaniem pełnej ostrożności. Należy dwukrotnie sprawdzić listę "nieaktualnych" pakietów i upewnić się, że Portage nie chce usunąć czegoś ważnego. W poniższym przykładzie dodajemy opcję -p, która wyświetli listę pakietów do usunięcia, bez ich usuwania.

Listing 8: Usuwanie niepotrzebnych pakietów

```
# emerge -p --depclean
```

Po zakończeniu depcleanu uruchamiamy revdep-rebuild, aby przebudować aplikacje, które mogą być połączone dynamicznie z usuniętymi bibliotekami. revdep-rebuild jest częścią pakietu gentoolkit.

Listing 9: Uruchomienie revdep-rebuild

```
# revdep-rebuild
```

Po zakończeniu tych wszystkich czynności system będzie używał nowych ustawień flag USE.

2.c. Zmienne USE specyficzne dla pakietów

Przeglądanie dostępnych flag USE

Weźmy na przykład mozillę i dowiedzmy się których flag USE używa. Użyjemy do tego polecenia emerge z parametrami --pretend oraz --verbose:

Listing 10: Przeglądanie używanych flag USE:

```
# emerge --pretend --verbose mozilla
These are the packages that I would merge, in order:

Calculating dependencies ...done!
[ebuild R ] www-client/mozilla-1.7.12-r2 USE="crypt gnome java mozsvg ssl
truetype xprint -debug -ipv6 -ldap -mozcalendar -mozdevelop -moznocompose
-moznoirc -moznomail -moznoxft -postgres -xinerama" 0 kB
```

emerge nie jest jedynym narzędziem wykorzystywanym w celu przeglądania informacji o pakietach. Do dyspozycji mamy jeszcze program equery, znajdujący się w pakiecie gentoolkit. Zaczniemy od zainstalowania gentoolkit:

Listing 11: Instalacja gentoolkit

```
# emerge gentoolkit
```

Następnie uruchamiamy equery z argumentem uses aby przejrzeć flagi USE dla konkretnego pakietu. Dla przykładu sprawdzmy pakiet gnumeric:

Listing 12: Użycie equery do przeglądania użytych flag USE:

```
# equery uses =gnumeric-1.6.3 -a
[ Searching for packages matching =gnumeric-1.6.3... ]
[ Colour Code : set unset ]
[ Legend       : Left column (U) - USE flags from make.conf   ]
[               : Right column (I) - USE flags packages was installed with ]
[ Found these USE variables for app-office/gnumeric-1.6.3 ]
  U I
  -- debug    : Tells configure and the makefiles to build for debugging.
                  Effects vary across packages, but generally it will at
                  least add -g to CFLAGS. Remember to set FEATURES=nostrip too
  -- gnome    : Adds GNOME support
++ python    : Adds support/bindings for the Python language
  -- static   : !!do not set this during bootstrap!! Causes binaries to be
                  statically linked instead of dynamically
```

3. Funkcje Portage

3.a. Funkcje Portage

Portage posiada szereg dodatkowych funkcji, które potrafią znacznie uprzyjemnić pracę z Gentoo. Wiele z nich opiera się na zewnętrznych programach, które zwiększą wydajność, stabilność i bezpieczeństwo pracy.

Aby włączyć lub wyłączyć określone dodatkowe funkcje Portage należy odpowiednio zmienić zmienną FEATURES w pliku /etc/make.conf. Zmienna ta to podzielona spacjami lista nazw dodatkowych możliwości. W niektórych przypadkach, aby móc korzystać z pewnych funkcji trzeba również zainstalować dodatkowe oprogramowanie.

Nie wszystkie funkcje, które Portage obsługuje są tutaj wymienione. By poznać wszystkie funkcje, należy przeczytać dokumentację make.conf:

Listing 1: Warto zjrzeć na stronę man pliku make.conf

```
$ man make.conf
```

By dowiedzieć się, jakie FEATURES są standardowo włączone, należy uruchomić emerge --info i poszukać zmiennej FEATURES za pomocą programu grep:

Listing 2: Sprawdzanie, czy FEATURES są już ustawione

```
$ emerge --info | grep FEATURES
```

3.b. DistCC

Czym jest DistCC?

Distcc to program, dzięki któremu możemy rozłożyć obciążenie związane z komplikacją pomiędzy kilka niekoniecznie identycznych maszyn. Klient distcc wysyła wszystkie potrzebne informacje do dostępnych serwerów DistCC (na których jest uruchomiony distccd), które następnie komplikują części kodu źródłowego dla klienta. Końcowym wynikiem jest krótszy czas komplikacji.

Dokładniejsze informacje na temat distcc (oraz informacje na temat tego, jak używać distcc w Gentoo) można odnaleźć Dokumentacji Distcc Gentoo.

Instalacja DistCC

Distcc jest dostarczany z graficznym monitorem, dzięki któremu możliwe jest obserwowanie postępu zadań, które komputer wysłał do serwerów distcc. Jeśli używany jest Gnome, należy umieścić "gnome" w ustawieniach flag USE. Jeśli nie jest zainstalowany Gnome, a mimo to chcielibyśmy mieć możliwość monitorowania distcc, należy umieścić w flagach USE "gtk".

Listing 3: Instalacja Distcc

```
# emerge distcc
```

Używanie distcc z Portage

Najpierw należy dodać distcc do zmiennej FEATURES w pliku /etc/make.conf. Następnie należy dostosować zmienną MAKEOPTS do swoich potrzeb. Zwykle ma ona postać -jX, gdzie X to liczba procesorów, na których uruchomiony jest distccd (włącznie z komputerem, na którym teraz pracujemy) powiększona o jeden. Czasem inne wartości od tych zalecanych przynoszą lepsze rezultaty.

Teraz trzeba uruchomić distcc-config i wprowadzić listę dostępnych serwerów DistCC. W naszym prostym przykładzie zakładamy, że dostępne serwery DistCC to: 192.168.1.102 (aktualny host), 192.168.1.103 i 192.168.1.104 (dwa "zdalne" hosty):

Listing 4: Użycie trzech serwerów DistCC

```
# distcc-config --set-hosts "192.168.1.102 192.168.1.103 192.168.1.104"
```

Trzeba też pamiętać o uruchomieniu demona distccd na zdalnych hostach:

Listing 5: Uruchamianie demona distcc

```
# rc-update add distccd default
# /etc/init.d/distccd start
```

3.c. ccache

Czym jest ccache?

ccache jest szybkim cache kompilatora. Dzięki niemu pliki pośrednie powstające w trakcie komplikacji będą cache'owane i podczas rekompilacji programu czas budowania plików wynikowych zostanie znacznie skrócony. W typowych sytuacjach czas komplikacji może być od 5 do 10 razy krótszy.

Szczegóły na temat ccache można odnaleźć na [stronie domowej ccache](#).

Instalacja ccache

Instalowanie ccache w Gentoo jest bardzo proste - jedyne, co należy zrobić, to zainstalować odpowiedni pakiet:

Listing 6: Instalacja ccache

```
# emerge ccache
```

Portage i ccache

Otwieramy plik `/etc/make.conf` i zmieniamy FEATURES tak, aby zawierało słowo kluczowe ccache oraz dodajemy zmienną CCACHE_SIZE o wartości "2G".

Listing 7: Zmiana CCACHE_SIZE w /etc/make.conf

```
CCACHE_SIZE="2G"
```

Aby sprawdzić czy ccache działa poprawnie, należy sprawdzić statystyki. Ponieważ Portage używa innych katalogów domowych ccache, należy ustawić zmienną CCACHE_DIR na początku polecenia:

Listing 8: Przeglądanie statystyk ccache

```
# CCACHE_DIR="/var/tmp/ccache" ccache -s
```

Katalog `/var/tmp/ccache` jest domyślną lokalizacją ccache w Portage. Jeżeli chcemy zmodyfikować tę pozycję należy ustawić zmienną CCACHE_DIR w pliku `/etc/make.conf`.

Jednak gdy będziemy uruchamiać polecenie ccache, będzie ono odwoływało się do domyślnej lokalizacji `${HOME}/.ccache`, dlatego też musimy za każdym razem ustawać zmienną CCACHE_DIR, gdy będziemy chcieli zobaczyć statystyki.

Używanie ccache dla komplikacji programów w C spoza Portage

Jeśli ccache ma być używane do komplikacji programów w C, ale nie znajdujących się w Portage, należy dodać katalog `/usr/lib/ccache/bin` na początku zmiennej PATH (przed wpisem `/usr/bin`). Robi się to edytując plik `/etc/profile`:

Listing 9: Edytowanie /etc/profile

```
PATH="/usr/lib/ccache/bin:${PATH}"
```

3.d. Pakiety binarne

Tworzenie pakietów binarnych

Portage umożliwia pracę z prekompilowanymi pakietami. Nie dostarczamy wprawdzie ich zestawów użytkownikom (poza GRP, które wychodzą co kilka miesięcy wraz z wydaniami Gentoo), ale mimo wszystko pozostawiamy możliwość korzystania z nich w naszym oprogramowaniu.

Jeśli dany pakiet już jest zainstalowany, można użyć polecenia quickpkg, które utworzy archiwum tar zawierające zainstalowane pliki (bardzo przydatne przy robieniu kopii zapasowych). Jeśli nie jest zainstalowany, należy skorzystać z polecenia emerge z opcją --buildpkg lub --buildpkgonly.

Aby Portage domyślnie tworzyło binarne pakiety, wystarczy umieścić słowo kluczowe buildpkg w zmiennej FEATURES.

Szersze możliwości budowania pakietów daje program catalyst. Wszystkie informacje o nim znajdują się na stronach [Catalyst Reference Manual](#) i w [Catalyst FAQ](#).

Instalacja prekompilowanych pakietów

Fakt, że Gentoo nie posiada repozytorium z prekompilowanymi pakietami nie oznacza, że użytkownicy nie mogą stworzyć takiego samodzielnie. Aby z niego korzystać, należy ustawić zmienną PORTAGE_BINHOST tak, aby na nie wskazywała. Na przykład, jeżeli prekompilowane pakiety znajdują się pod adresem ftp://buildhost/gentoo:

Listing 10: Konfiguracja zmiennej PORTAGE_BINHOST w pliku /etc/make.conf

```
PORTAGE_BINHOST="ftp://buildhost/gentoo"
```

Za każdym razem, gdy chcemy zainstalować prekompilowany pakiet, musimy skorzystać z parametru --getbinpkg razem z opcją --usepkg. Pierwsza opcja nakazuje pobrać prekompilowany pakiet ze zdalnego serwera, druga nakazuje emerge skorzystanie ze sciagniętego pakietu podczas instalacji.

Na przykład, aby zainstalować gnumeric z prekompilowanego pakietu:

Listing 11: Instalacja prekompilowanego gnumeric

```
# emerge --usepkg --getbinpkg gnumeric
```

Więcej informacji o prekompilowanych pakietach znajduje się na stronie man programu emerge.

Listing 12: Strona man programu emerge

```
$ man emerge
```

4. Skrypty startowe

4.a. Poziomy działania

Uruchamianie systemu operacyjnego

Podczas uruchamiania systemu operacyjnego na ekranie pojawia się dużo, nie zawsze zrozumiałego tekstu. Gdy przyjrzeć się dokładniej, można zauważać, że tekst ten jest za każdym razem taki sam. Cały ten proces nazywamy sekwencją startową, która (w większym lub mniejszym stopniu) jest skonfigurowana statycznie.

Najpierw bootloader ładuje obraz jądra systemu do pamięci i zleca procesorowi jego wykonanie. W chwili kiedy jądro zostanie załadowane i wykonane, uruchamiane są specyficzne zadania, związane ściśle z jądrem po czym uruchamiany jest proces init.

Proces ten następnie upewnia się czy wszystkie systemy plików (zdefiniowane w /etc/fstab) zostały poprawnie zamontowane i są gotowe do pracy. Następnie uruchamiane są poszczególne skrypty umieszczone w katalogu /etc/init.d, które mają za zadanie uruchomić kolejno wszystkie usługi niezbędne do poprawnego działania systemu.

Na koniec, kiedy wszystkie skrypty zostaną wykonane, init aktywuje terminale (w większości przypadków są to po prostu wirtualne konsole, między którymi można się przełączać za pomocą kombinacji klawiszy Alt-F1, Alt-F2 itd.) przy pomocy służącego do tego programu pod nazwą getty. Sprawdza on czy użytkownik może się zalogować na dany terminal, uruchamiając login.

Skrypty Init

Skrypty umieszczone w katalogu /etc/init.d nie są uruchamiane przez init w przypadkowej kolejności. Co więcej, nie są uruchamiane wszystkie naraz lecz w określonej kolejności. Informacje na ten temat tej kolejności pobierane są z katalogu /etc/runlevels.

Na samym początku init inicjuje te skrypty, do których dowiązania symboliczne znajdują się w katalogu /etc/runlevels/boot. Zazwyczaj uruchamiane są one w kolejności alfabetycznej. Wyjątek stanowią te, które posiadają informacje o zależnościach. Mówią one o tym, że do prawidłowego działania danej usługi musi wcześniej zostać uruchomiona inna.

Kiedy skrypty mające dowiązanie w /etc/runlevels/boot zostaną uruchomione, init kontynuuje uruchamianie tych, do których dowiązania znajdują się w katalogu /etc/runlevels/default. Podobnie jak w poprzednim przypadku, uruchamiane są w kolejności alfabetycznej. Wyjątek stanowią tylko sytuacje, gdy muszą zostać spełnione zależności niezbędne do poprawnego przeprowadzenia procesu startowego.

Jak działa Init

Oczywiście o wszystkim nie decyduje sam init. Potrzebuje on stosownego pliku konfiguracyjnego, który zawiera informacje o zadaniach jakie ma wykonać. Ten plik to /etc/inittab.

Na początku tej części dokumentu jest wzmianka o tym, że init w początkowej fazie działania sprawdza czy systemy plików zostały zamontowane poprawnie. Definicja tego zadania w /etc/inittab wygląda następująco:

Listing 1: Inicjacja systemu w /etc/inittab

```
si::sysinit:/sbin/rc sysinit
```

Powyższa linia mówi procesowi init, że w celu inicjacji systemu ma wykonać polecenie /sbin/rc sysinit. Tak naprawdę to właśnie skrypt /sbin/rc zajmuje się inicjacją, a init jedynie zleca zadania innym procesom.

Następnie init uruchamia wszystkie skrypty, do których dowiązania symboliczne znajdują się we wspomnianym wcześniej katalogu /etc/runlevels/boot. W pliku konfiguracyjnym jest to zdefiniowane w następujący sposób:

Listing 2: Kontynuacja procesu uruchamiania systemu

```
rc::bootwait:/sbin/rc boot
```

Ponownie skrypt rc wykonuje niezbędne zadania. Argument dla rc (boot) jest taki sam jak nazwa podkatalogu w /etc/runlevels, w którym znajdują się dowiązania do skryptów wykonywanych w tej części procesu uruchamiania systemu.

Następnie init sprawdza plik /etc/inittab w celu odszukania informacji, w który poziom działania (runlevel) ma "wejść" system:

Listing 3: Linia initdefault

```
id:3:initdefault:
```

W tym przypadku jest to poziom (runlevel) o numerze 3. Dzięki tej informacji init sprawdza co musi zostać uruchomione aby system zaczął działać w trzecim poziomie (rulevelu 3).

Listing 4: Definicja poziomów działania

```
10:0:wait:/sbin/rc shutdown
13:3:wait:/sbin/rc default
14:4:wait:/sbin/rc default
15:5:wait:/sbin/rc default
16:6:wait:/sbin/rc reboot
```

W linii, która definiuje trzeci poziom, podobnie jak w przypadku poprzednich, jest odwołanie do skryptu rc. Tym razem jest on uruchamiany z argumentem default. Argument ten brzmi tak samo, jak nazwa jednego z podkatalogów w /etc/runlevels.

Po tym jak rc zakończy swoją pracę, init decyduje o tym jakie, oraz przy użyciu jakich poleceń, mają zostać aktywowane wirtualne konsole.

Listing 5: Definicja konsol wirtualnych

```
c1:12345:respawn:/sbin/agetty 38400 tty1 linux
c2:12345:respawn:/sbin/agetty 38400 tty2 linux
c3:12345:respawn:/sbin/agetty 38400 tty3 linux
c4:12345:respawn:/sbin/agetty 38400 tty4 linux
c5:12345:respawn:/sbin/agetty 38400 tty5 linux
c6:12345:respawn:/sbin/agetty 38400 tty6 linux
```

Co to jest poziom działania (runlevel)?

Init używając notacji, w której każdy poziom działania ma swój numer decyduje o tym, który z nich ma być w danej chwili aktywny. Poziom działania (runlevel) to stan, w którym uruchomiony jest system operacyjny. Każdy z poziomów charakteryzuje się pewnym zestawem skryptów, które muszą być wykonane podczas wchodzenia lub wychodzenia z danego poziomu.

Gentoo posiada siedem zdefiniowanych poziomów: trzy wewnętrzne i cztery definiowane przez użytkownika. Wewnętrzne nazywają się sysinit, shutdown i reboot. Jak nietrudno się domyślić służą one kolejno do inicjacji, wyłączania oraz ponownego uruchamiania systemu.

Poziomy definiowane przez użytkownika związane są z podkatalogami /etc/runlevels: boot, default, nonetwork i single. W poziomie boot uruchamiane są wszystkie niezbędne usługi systemowe używane w pozostałych poziomach. Pozostałe trzy różnią się rodzajem uruchamianych usług: default służy do uruchamiania "standardowych" operacji, nonetwork wykorzystywany jest w przypadkach kiedy do uruchomienia danej usługi nie jest wymagane połączenie z siecią, zaś single używany jest tylko wtedy, gdy system wymaga naprawy.

Praca ze skryptami Init

Skrypty uruchamiane przez proces rc nazywane są skryptami init (ang. init scripts). Umieszczone są w katalogu /etc/init.d i mogą być uruchamiane wraz z następującymi argumentami: start, stop, restart, pause, zap, status, ineed, iuse, needsme, usesme lub broken.

Aby uruchomić, zatrzymać lub przeładować dowolną usługę (wraz z powiązanymi z nią innymi usługami) należy użyć odpowiednio start, stop i restart. Przykładowo:

Listing 6: Uruchamianie Postfixa

```
# /etc/init.d/postfix start
```

Uwaga: Wyłączane lub przeładowywane są tylko te usługi,, które tego wymagają. Inne, które są powiązane z przeładowaną usługą, jeśli nie ma takiej potrzeby, nie są restartowane.

Aby wyłączyć daną usługę pozostawiając przy życiu usługi z nią powiązane należy użyć argumentu pause:

Listing 7: Wyłączanie Postfixa, zachowując włączone powiązane z nim usługi

```
# /etc/init.d/postfix pause
```

Aby zobaczyć jaki status ma aktualnie dana usługa (włączony, wyłączony...) trzeba użyć argumentu status:

Listing 8: Informacje o statusie postfixa

```
# /etc/init.d/postfix status
```

Jeśli powyższe polecenie zwróci informację, że Postfix jest uruchomiony, lecz faktycznie będzie inaczej, należy użyć argumentu zap w celu uaktualnienia informacji o statusie.

Listing 9: Uaktualnienie informacji o statusie postfixa

```
# /etc/init.d/postfix zap
```

Do sprawdzenia jakie zależności posiada usługa trzeba użyć argumentu iuse lub ineed. Dzięki ineed można uzyskać listę tych, które są niezbędne do prawidłowego działania danej usługi. iuse z kolei pokazuje te usługi, które mogą być używane lecz nie są niezbędne do jego uruchomienia i poprawnego funkcjonowania.

Listing 10: Zapytanie o listę usług niezbędnych do działania Postfixa

```
# /etc/init.d/postfix ineed
```

Podobnie można zapytać, które z usług w systemie wymagają danej usługi (needsme) lub mogą lecz nie muszą go używać (usesme):

Listing 11: Zapytanie o listę usług, które wymagają Postfixa

```
# /etc/init.d/postfix needsme
```

Ostatnią z możliwości jest użycie argumentu, który wyświetli listę brakujących z listy wymaganych usług.

Listing 12: Zapytanie o listę brakujących usług powiązanych z Postfixem

```
# /etc/init.d/postfix broken
```

4.b. Praca z rc-update

Co to jest rc-update?

W celu ustalenia poprawnej kolejności uruchamiania usług system init w Gentoo korzysta z drzewa zależności. Utrzymanie i zarządzanie takim drzewem bez użycia dodatkowych narzędzi byłoby bardzo nudnym i stosunkowo trudnym zadaniem. Na szczęście w Gentoo są już gotowe narzędzia, które znacznie ułatwiają zarządzanie poziomami działania oraz skryptami init.

Przy pomocy rc-update można dodawać i usuwać skrypty init z poziomu działania (runlevela). rc-update za każdym razem zleca skryptowi depscan.sh odbudowanie na nowo wspomnianego drzewa zależności.

Dodawanie i usuwanie usług

Pierwsze usługi są dodawane do poziomów działania już podczas procesu instalacyjnego. Wówczas można było nie skojarzyć czym jest na przykład poziom o nazwie "default", teraz powinno to być jasne. W celu dodania lub usunięcia usługi, rc-update wymaga podania między innymi argumentu określającego akcję (co rc-update ma zrobić): add, del lub show.

Zatem w celu dodania lub usunięcia skryptu init, należy wykonać polecenie rc-update wraz z argumentami add lub del, podając dalej nazwę skryptu oraz poziomu. Na przykład:

Listing 13: Usuwanie Postfixa z poziomu default

```
# rc-update del postfix default
```

Polecenie rc-update pokazuje listę wszystkich dostępnych skryptów wraz z informacją w którym z poziomów są one uruchamiane:

Listing 14: Informacje o dostępnych skryptach init

```
# rc-update show
```

4.c. Konfiguracja usług

Dlaczego dodatkowa konfiguracja jest potrzebna?

Skrypty init mogą być niekiedy dość skomplikowane. Dlatego część użytkowników nie jest zbytnio zainteresowana ich edytowaniem i modyfikacją z uwagi na możliwość popełnienia błędów. Jednak czasami możliwość zmiany konfiguracji usługi jest bardzo ważna. Na przykład w momencie kiedy zaistnieje potrzeba samodzielne dodania jakiejś opcji.

Drugim powodem, dla którego ingerencja w skrypty init może okazać się pomocna jest możliwość aktualnienia skryptów bez obawy przed tym, że dokonane zmiany nie zostaną zastosowane.

Katalog /etc/conf.d

Gentoo umożliwia bardzo prosty sposób konfiguracji poszczególnych usług. Każdy skrypt init może być skonfigurowany za pomocą stosownego pliku w katalogu /etc/conf.d. Na przykład skrypt apache2 (/etc/init.d/apache2) posiada swój własny plik konfiguracyjny, /etc/conf.d/apache2, w którym można umieścić wszelkie opcje z jakimi ma się uruchomić serwer Apache 2:

Listing 15: Zmienna zdefiniowana w pliku /etc/conf.d/apache2

```
APACHE2_OPTS="-D PHP4"
```

Plik konfiguracyjny zawiera zmienne (podobnie jak /etc/make.conf), czyniąc konfigurację serwisów bardzo łatwą. Dostarcza nam to także więcej informacji na temat zmiennych (jako komentarz).

4.d. Pisanie skryptów Init

Czy muszę to robić?

Nie. Zwykle umiejętność pisania skryptów dla inita nie jest wymaganą umiejętnością ponieważ wraz z dystrybucją Gentoo dostarczane są wszystkie niezbędne skrypty, które pozwalają na uruchamianie wszystkich usług. Aczkolwiek umiejętność ta może okazać się przydatna, kiedy zainstalowana zostanie w systemie usługa, bez użycia do tego Portage. Wówczas będzie trzeba napisać skrypt samodzielnie.

Nie można używać skryptów init, które nie są napisane specjalnie dla Gentoo: skrypty init w Gentoo nie są kompatybilne ze skryptami z innych dystrybucji!

Szablon

Poniżej znajduje się szablon skryptu init.

Listing 16: Szablon skryptu init

```
#!/sbin/runscript

depend() {
    (Informacje o zależnościach)
}

start() {
    (Komendy niezbędne do uruchomienia usługi)
}

stop() {
    (Komendy niezbędne do jej wyłączenia)
}

restart() {
    (Polecenia służące do restartu usługi)
}
```

Każdy skrypt wymaga zdefiniowanej funkcji start(). Pozostałe funkcje są opcjonalne.

Zależności

W tym miejscu można zdefiniować dwa rodzaje zależności: use i need. Wspomniane wcześniej need są bardziej restrykcyjne niż zależności zdefiniowane jako use. Należy wybrać i dodać tu stosowne usługi, od których zależna będzie ta, dla której piszemy skrypt. Można też zdefiniować zależności wirtualne.

Zależności wirtualne to takie zależności, w których nie określą się ściśle konkretnej usługi. Przykładowo skrypt init wymaga działającego systemu logowania, lecz nie jest jasno określone jakiego. W Gentoo dostępnych jest kilka systemów logowania (metalogd, syslog-ng, sysklogd, ...). Zdefiniowanie każdego z nich (zainstalowanie i uruchomienie wszystkich wymienionych wyżej systemów logowania nie wydaje się być najlepszym pomysłem) nie było by dobrym rozwiązaniem. Jak można się jednak przekonać wszystkie z tych usług są akceptowane dzięki zależnościom wirtualnym.

Rzućmy okiem na informacje o zależnościach dla usługi Postfix.

Listing 17: Zależności Postfixa

```
depend() {
    need net
    use logger dns
    provide mta
}
```

Jak widać, postfix:

- wymaga usługi net (jest to zależność wirtualna, która może być spełniona przykładowo przez /etc/init.d/net.eth0).
- współpracuje z usługą logger (jest to zależność wirtualna, którą spełnia przykładowo /etc/init.d/syslog-ng).
- współpracuje z usługą dns (zależność wirtualna, którą spełnia przykładowo /etc/init.d/named).
- zapewnia usługę mta (zależność wirtualna, którą spełniają wszystkie serwery pocztowe).

Kontrola kolejności

Czasami nie potrzeba osobnego skryptu inicjującego. Chcemy jednak, aby usługa była uruchamiana przed (lub po) uruchomieniem innej usługi, jeśli ta jest dostępna w systemie i uruchamia się na tym samym poziomie działania. Informacji na ten temat możesz dostarczyć skryptowi za pomocą opcji before lub after.

Przyjrzyjmy się bliżej ustawieniom usługi Portmap:

Listing 18: Funkcja depend() usługi Portmap

```
depend() {
    need net
    before inetd
    before xinetd
}
```

Można użyć znaku "*" aby objąć wszystkie usługi w tym samym poziomie działania, nie jest to jednak zalecana metoda.

Listing 19: Uruchamianie skryptu init jako pierwszego w poziomie działania

```
depend() {
    before *
}
```

Funkcje standardowe

Do tego aby funkcja depend() spełniała swoje zadanie, potrzebna jest poprawna definicja funkcji start(). Funkcja ta zawiera polecenia niezbędne do uruchomienia usługi.

Wskazane jest użycie opcji ebegin i eend, dzięki którym można poinformować użytkownika co się w danym momencie dzieje:

Listing 20: Przykład funkcji start()

```
start() {
    ebegin "Uruchamiam moja_usługa"
    start-stop-daemon --start --quiet --exec /path/to/moja_usługa
    eend $?
}
```

Więcej przykładów użycia funkcji start() można odnaleźć w kodzie źródłowym dostępnych w systemie skryptów init, które znajdują się w katalogu /etc/init.d. Jeśli chcemy zdobyć więcej informacji start-stop-daemon posiada znakomitą dokumentację, która dostępna jest w formacie podręcznika systemowego:

Listing 21: Uzyskiwanie informacji o start-stop-daemon

```
# man start-stop-daemon
```

Pozostałe funkcje jakie można definiować to: stop() i restart(). Nie są one jednak konieczne! System init jest dostatecznie inteligentny aby poradzić sobie z ich brakiem dzięki start-stop-daemon.

Składnia skryptów startowych Gentoo opierają się na bashu przez co można w nich używać instrukcji zgodnych z bashem.

Dodawanie niestandardowych opcji

Jeśli chcemy aby init posiadał więcej opcji niż te dotychczas omówione, należy dodać nową opcję do zmiennej opts i stworzyć funkcję o takiej samej nazwie. Na przykład, aby utworzyć opcję o nazwie restartdelay:

Listing 22: Dodanie opcji restartdelay

```
opts="${opts} restartdelay"

restartdelay() {
    stop()
    sleep 3      # czekaj 3 sekundy przed ponownym uruchomieniem
    start()
}
```

Zmienne konfiguracyjne dla usług

Aby skrypt uruchamiający daną usługę sięgał do plików konfiguracyjnych, zlokalizowanych w /etc/conf.d nie trzeba praktycznie robić niczego. W chwili kiedy skrypt zostanie uruchomiony, przetworzone zostaną następujące pliki:

- /etc/conf.d/<nasz skrypt init>
- /etc/conf.d/basic
- /etc/rc.conf

Jeśli skrypt zawiera jakieś zależności wirtualne (np. takie jak net), pliki związane z tymi zależnościami (w tym przypadku /etc/conf.d/net) także zostaną przetworzone.

4.e. Zmiana zachowania poziomu działania

Kto może mieć z tego korzyści?

Wielu użytkowników laptopów zna taką sytuację: dopiero po powrocie do domu chcą uruchomić net.eth0, gdyż gdy są w drodze, to i tak nie ma sensu go uruchamiać, bo i tak nie ma dostępu do sieci. W Gentoo można dowolnie modyfikować zachowanie poziomów działania.

Możemy, na przykład utworzyć drugi "domyślny" poziom, który używa innych skryptów startowych. Można wybierać, którego poziomu działania chce się używać podczas startu systemu.

Używanie softlevela

Po pierwsze, trzeba utworzyć katalog poziomów działania dla swojego drugiego "domyślnego" poziomu działania.

Jako przykład stworzymy katalog offline:

Listing 23: Tworzenie katalogu poziomu działania

```
# mkdir /etc/runlevels/offline
```

Należy dodać potrzebne skrypty startowe do nowo utworzonego katalogu. Na przykład, jeżeli chcemy mieć dokładną kopię aktualnego domyślnego poziomu działania, wyłączając net.eth0:

Listing 24: Dodawanie potrzebnych skryptów startowych

```
(Kopiowanie wszystkich usług z domyślnego runlevela do runlevela offline
# cd /etc/runlevels/default
# for service in *; do rc-update add $service offline; done
(Usuwanie niechcianej usługi z runlevela offline
# rc-update del net.eth0 offline
(Wyświetlenie wszystkich aktywnych usług dla runlevela offline
# rc-update show offline
(Częściowe wyjście)
          acpid | offline
          domainname | offline
          local | offline
          net.eth0 |
```

Teraz należy wyedytować pliki konfiguracyjne bootloadera i dodać wpis dla poziomu działania offline. Dla przykładu w /boot/grub/grub.conf:

Listing 25: Dodawanie wpisu dla poziomu działania offline

```
title Gentoo Linux Tryb Offline
root (hd0,0)
kernel (hd0,0)/kernel-2.4.25 root=/dev/hda3 softlevel=offline
```

Teraz już jest wszystko ustalone. Jeżeli wystartowany zostanie system i wybrana zostanie dodana przed chwilą pozycja, zamiast domyślnego poziomu działania będzie używany poziom offline.

Używanie bootlevela

Używanie bootlevela jest analogiczne do softlevela. Jedyną różnicą jest definiowanie drugiego "rozruchowego" poziomu uruchamiania zamiast drugiego "domyślnego" poziomu uruchamiania.

5. Zmienne środowiskowe

5.a. Zmienne środowiskowe

Czym są zmienne środowiskowe?

Zmienne środowiskowe to nazwa obiektów zawierających informacje, które używane są przez jeden lub wiele programów w systemie operacyjnym. Wielu użytkowników (zwłaszcza mających styczność z Linusem od niedawna) traktuje je jako coś nie do ogarnięcia. Nic bardziej mylnego! Dzięki zmiennym środowiskowym zmiana konfiguracji jednego lub kilku programów jest banalnie prosta.

Przykłady ważniejszych zmiennych środowiskowych

Poniższa tabela przedstawia listę ważniejszych zmiennych używanych przez system Linux, wraz z krótkim opisem. Przykładowe ich wartości znajdują się pod tabelą.

Zmienna	Opis
PATH	Ta zmienna zawiera oddzieloną dwukropkami listę katalogów, w których system operacyjny szuka plików z prawami do uruchomienia. Jeśli w konsoli wpiszesz nazwę programu mającego prawa do uruchamiania (np. ls, rc-update lub emerge) lecz program ten nie znajduje się w jednym z katalogów zdefiniowanych w zmiennej PATH, system nie wykona tego programu (chyba, że wpiszesz pełną ścieżkę do miejsca gdzie znajduje się ten program, np. /bin/ls).
ROOTPATH	Ta zmienna spełnia prawie taką samą funkcję jak PATH, z tą tylko różnicą, że zawiera informacje o katalogach które są sprawdzane w poszukiwaniu programów dla superużytkownika (czyli root'a).
LDPATH	Zmienna zawiera podzieloną dwukropkami listę katalogów, które konsolidator przeszukuje w celu odnalezienia bibliotek.
MANPATH	Zmienna, podobnie jak inne zawiera listę katalogów oddzielonych dwukropkiem, w których man szukał będzie dokumentów w odpowiednim dla siebie formacie.
INFODIR	Zmienna ta, to lista katalogów oddzielona znakiem dwukropka, które przeszukiwane są przez program info w celu odnalezienia dokumentacji w odpowiednim dla niego formacie.
PAGER	Zmienna zawiera ścieżkę do programu, który służy do prezentacji zawartości plików (przykładowo less lub more)
EDITOR	Zmienna zawiera ścieżkę do programu używanego do edycji plików (przykładowo nano lub vi)
KDEDIRS	Zmienna zawiera listę oddzielonych znakiem dwukropka katalogów, w których mieszczą się materiały związane z środowiskiem graficznym KDE.
CLASSPATH	Zmienna zawiera listę katalogów oddzielonych znakiem dwukropka, w których znajdują się klasy języka Java.
CONFIG_PROTECT	Zmienna ta zawiera listę oddzielonych znakiem spacji katalogów, które mają być zabezpieczone przez Portage w trakcie dokonywania aktualnień oprogramowania.
CONFIG_PROTECT_MASK	Zmienna zawiera listę oddzielonych znakiem spacji katalogów, które nie mają być zabezpieczane przez Portage podczas dokonywania aktualnienia oprogramowania.

Poniżej znajdują się przykładowe wartości omówionych zmiennych środowiskowych:

Listing 1: Przykładowe wartości zmiennych

```
PATH="/bin:/usr/bin:/usr/local/bin:/opt/bin:/usr/games/bin"
ROOTPATH="/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/sbin:/usr/local/bin"
LDPATH="/lib:/usr/lib:/usr/local/lib:/usr/lib/gcc-lib/i686-pc-linux-gnu/3.2.3"
MANPATH="/usr/share/man:/usr/local/share/man"
INFODIR="/usr/share/info:/usr/local/share/info"
PAGER="/usr/bin/less"
EDITOR="/usr/bin/vim"
KDEDIRS="/usr"
CLASSPATH="/opt/blackdown-jre-1.4.1/lib/rt.jar::"
CONFIG_PROTECT="/usr/X11R6/lib/X11/xkb /opt/tomcat/conf \
    /usr/kde/3.1/share/config /usr/share/texmf/tex/generic/config/ \
```

```
/usr/share/texmf/tex/latex/config/ /usr/share/config"
CONFIG_PROTECT_MASK="/etc/gconf"
```

5.b. Definiowanie zmiennych globalnych

Katalog /etc/env.d

Aby skupić w jednym miejscu definicje zmiennych, w Gentoo wprowadzono katalog /etc/env.d. W katalogu tym odnaleźć można pliki takie jak 00basic, 05gcc i inne. Zawierają one zmienne potrzebne do działania programów, których nazwy najczęściej są takie jak nazwy plików znajdujących się w katalogu /etc/env.d.

Na przykład po zainstalowaniu gcc, w katalogu /etc/env.d tworzony jest plik o nazwie 05gcc, który zawiera definicje następujących zmiennych:

Listing 2: /etc/env.d/05gcc

```
PATH="/usr/i686-pc-linux-gnu/gcc-bin/3.2"
ROOTPATH="/usr/i686-pc-linux-gnu/gcc-bin/3.2"
MANPATH="/usr/share/gcc-data/i686-pc-linux-gnu/3.2/man"
INFOPATH="/usr/share/gcc-data/i686-pc-linux-gnu/3.2/info"
CC="gcc"
CXX="g++"
LDPATH="/usr/lib/gcc-lib/i686-pc-linux-gnu/3.2.3"
```

W innych dystrybucjach trzeba by zmienić lub dodać definicje powyższych zmiennych systemowych do pliku /etc/profile lub w innym miejscu. W Gentoo jest to znacznie prostsze i nie wymaga w zasadzie ingerencji użytkownika.

W przypadku, kiedy gcc jest uaktualniane, uaktualniany jest także plik /etc/env.d/05gcc. Dzieje się to bez udziału użytkownika.

Rozwiążanie to przynosi korzyści nie tylko dla systemu Portage, ale także dla użytkownika. Sporadycznie użytkownik może być zapytany o to, jaką wartość ma mieć pewna zmienna środowiskowa. Może to mieć miejsce w przypadku zmiennej http_proxy. Zamiast zamiast ingerować w plik /etc/profile, można utworzyć plik (/etc/env.d/99local) i wpisać tam definicje swoich zmiennych. Na przykład:

Listing 3: /etc/env.d/99local

```
http_proxy="proxy.server.com:8080"
```

Dzięki takiemu sposobowi definiowania zmiennych środowiskowych masz szybki wgląd na w te zmienne, które zdefiniowałeś samodzielnie.

Skrypt env-update

Kilka plików w katalogu /etc/env.d definiuje zmienną PATH. Nie jest to błędem. Kiedy wykonane zostanie env-update, to skrypt ten doda do siebie wszystkie definicje tej samej zmiennej, uaktualniając na koniec zmienne środowiskowe w systemie. Daje to możliwość dodawania własnych zmiennych środowiskowych bez obawy przed tym, że zmodyfikowana zostanie już istniejąca zmienna (pozbawiając ją dotychczasowej wartości).

Skrypt env-update dodaje wartości zmiennych w kolejności alfabetycznej. Tłumaczy to fakt dlaczego nazwy plików w katalogu /etc/env.d zaczynają się od dwóch cyfr.

Listing 4: Kolejność w jakiej env-update uaktualnia zmienne

00basic	99kde-env	99local
-----+	-----+	-----+

```
PATH="/bin:/usr/bin:/usr/kde/3.2/bin:/usr/local/bin"
```

Tylko wartości niektórych zmiennych są ze sobą łączone, należą do nich: KDEDIRS, PATH, CLASSPATH, LDPATH, MANPATH, INFODIR, INFOPATH, ROOTPATH, CONFIG_PROTECT, CONFIG_PROTECT_MASK, PRELINK_PATH i PRELINK_PATH_MASK. Dla wszystkich pozostałych zmiennych ważna jest ostatnia zdefiniowana wartość (pliki w kolejności alfabetycznej w katalogu /etc/env.d).

Uruchomienie skryptu env-update powoduje utworzenie zmiennych systemowych i umieszczenie ich w pliku /etc/profile.env (z którego korzysta /etc/profile). Skrypt ten pobiera także informacje z (opisanej wcześniej) zmiennej LDPATH i używa ich do utworzenia pliku /etc/ld.so.conf. Po wykonaniu tej czynności uruchamia polecenie ldconfig w celu odbudowania wspomnianego pliku /etc/ld.so.cache, który jest używany przez konsolidator.

Jeśli chcemy zaobserwować efekt, jaki przynosi wykonanie polecenia env-update, należy uruchomić następujące polecenia w celu uaktualnienia zmiennych środowiskowych. Użytkownicy mający już pierwszą instalację Gentoo za sobą pewnie pamiętają polecenie:

Listing 5: Uaktualnianie zmiennych środowiskowych

```
# env-update && source /etc/profile
```

Uwaga: Powyższe polecenie uaktualnia zmienne tylko na terminalu, na którym zostało uruchomione i jego terminalach potomnych. W związku z tym jeśli pracuje się w X11 należy albo wpisywać source /etc/profile w każdym terminalu, który się otwiera albo zrestartować serwer X tak żeby wszystkie nowe terminale posiadały nowe zmienne. Użytkownicy menedżerów logowania powinni przełączyć się na konto roota i wpisać /etc/init.d/xdm restart. Jeśli się tego nie zrobi to konieczne będzie przelogowanie się w X by uruchamić terminale potomne z nowymi zmiennymi.

5.c. Definiowanie zmiennych lokalnych

Zmienne użytkownika

Nie zawsze użytkownik chce aby zdefiniowane przez niego zmienne miały charakter globalny (były dostępne dla innych użytkowników w systemie). Na przykład może zechcieć dodać katalog /home/moj_uzytkownik/bin i katalog, w którym obecnie się znajduje do zmiennej PATH, lecz nie chce aby była ona dostępna dla pozostałych. Aby zdefiniować taką zmienną środowiskową lokalną, do pliku ~/.bashrc lub ~/.bash_profile należy dodać następującą linię:

Listing 6: Definicja lokalnej zmiennej PATH w pliku ~/.bashrc

```
Dwukropki na końcu zmiennej oznaczają, że wejdzie do niej także katalog, w którym będzie przebywała
PATH="${PATH}:/home/my_user/bin:"
```

Po wylogowaniu się i ponownym zalogowaniu, wartość zmiennej PATH będzie uaktualniona.

Zmienne bieżącej sesji

W niektórych przypadkach przydatna jest możliwość definiowania zmiennych, które używane są tylko w trakcie trwania bieżącej sesji. Na przykład może pojawić się potrzeba używania programów z katalogu tymczasowego, który nie jest ujęty standardowo w zmiennej PATH.

W tym przypadku można po prostu zdefiniować wartość tej zmiennej za pomocą polecenia export. Tak zdefiniowana zmienna będzie miała swoją wartość do momentu wylogowania się.

Listing 7: Definiowanie zmiennych środowiskowych dla bieżącej sesji

```
# export PATH="${PATH}:/home/my_user/tmp/usr/bin"
```

C. Praca z Portage

1. Pliki i katalogi

1.a. Pliki Portage

Dyrektywy konfiguracji

Domyślna konfiguracja Portage znajduje się w pliku `/etc/make.globals`. Gdy mu się przyjrzymy możemy zauważyc, że Portage jest konfigurowane za pomocą zmiennych. Znaczenie poszczególnych zmiennych omówimy w dalszych rozdziałach.

Portage ma również domyślne pliki konfiguracyjne wewnątrz wybranego profilu: `/etc/make.profile/make.defaults` ponieważ większość dyrektyw konfiguracji zależy od architektury. Wybrany profil jest zdefiniowany przez dowiązanie symboliczne `/etc/make.profile`. Cała konfiguracja Portage znajduje się w pliku profilu oraz w plikach profili nadzędnych. Więcej informacji o profilach i katalogu `/etc/make.profile` znajduje się w dalszych częściach tego Podręcznika.

Nie należy edytować plików `make.globals` ani `make.defaults` w celu zmiany jakiegokolwiek znajdującej się w nich zmiennej. Zamiast tego powinno się skorzystać z pliku `/etc/make.conf`, który jest pozycją nadzędną nad wyżej wymienionymi plikami i jest jedynym odpowiednim miejscem do wprowadzania jakiegokolwiek zmian do konfiguracji. Jeśli brakuje inwencji i nie ma się pomysłu co wpisać do tego pliku, warto zapoznać się z przykładowym plikiem `/etc/make.conf.example`.

Istnieje również możliwość zdefiniowania zmiennej konfiguracyjnej Portage jako zmiennej środowiskowej, ale nie jest to zalecana metoda.

Informacje specyficzne dla profilu

Wspominaliśmy już o katalogu `/etc/make.profile`. Nie jest to de facto katalog, lecz symboliczne dowiązanie do katalogu profilu znajdującego się wewnątrz `/usr/portage/profiles`. Profile mogą znajdować się w dowolnym miejscu na dysku, wystarczy, że to dowiązanie wskazuje na prawidłowy katalog.

Każdy profil zawiera informacje specyficzne dla danej architektury. Należą do nich między innymi lista pakietów niezbędnych dla prawidłowego działania systemu oraz lista pakietów niedziałających (bądź zamaskowanych) na danym systemie.

Konfiguracja specyficzna dla użytkownika

Aby zmienić związane z instalacją pakietów zachowanie Portage, należy udać się do katalogu `/etc/portage`. Polecamy wpisywanie tam całej własnej konfiguracji, nalegamy też na rezygnowanie z konfigurowania Portage przez zewnętrzne zmienne środowiskowe.

Wewnątrz `/etc/portage` można stworzyć następujące pliki:

- `package.mask`, w którym znajduje się lista pakietów, których nie chcemy instalować.
- `package.unmask`, w którym znajduje się lista pakietów, które mają być instalowane wbrew zaleceniom developerów.
- `package.keywords`, w którym znajduje się lista pakietów, które zamierza się zainstalować pomimo faktu, że nie są do końca kompatybilne z danym systemem bądź architekturą.
- `package.use`, w którym znajduje się lista flag USE, których chce się używać dla określonych pakietów, a które różnią się od tych ustawionych globalnie w systemie.

Więcej informacji o katalogu `/etc/portage` oraz pełna lista plików, które można tam stworzyć, znajduje się w manualu Portage:

Listing 1: Czytanie strony man dla Portage

```
$ man portage
```

Zmiana lokalizacji plików i katalogów należących do Portage

Omówione powyżej pliki zawsze muszą znajdować się w tym samym, określonym miejscu, gdyż tylko tam Portage będzie ich szukało. Można jednak zmienić lokalizację innych katalogów używanych przez system, takich jak na przykład miejsce zapisywania kodu źródłowego, katalog, w którym budowane są programy, czy miejsce, w którym znajduje się drzewo Portage.

Ścieżki do powyższych miejsc są doskonale znane wszystkim użytkownikom Gentoo. Jeśli jednak z jakichś względów zamierza się je zmienić można to zrobić poprzez plik `/etc/make.conf`. W pozostałej części tego rozdziału omówimy wszystkie specjalne lokalizacje w jakich działa Portage oraz sposoby ich zmianiania.

Wszystkie zawarte w tym dokumencie informacje można uzyskać czytając manuale Portage i `make.conf`.

1.b. Zapisywanie plików

Drzewo Portage

Domyślnie drzewo Portage znajduje się w katalogu `/usr/portage`, który definiowany jest przez zmienną `PORTEXEC`. Po zmianie wartości tej zmiennej należy pamiętać również o wprowadzeniu odpowiednich zmian w `/etc/make.profile`.

Jeśli zmodyfikuje się zmienną `PORTEXEC` to należy poprawić też zmienne `PKGDIR`, `DISTDIR` i `RPMDIR`, gdyż programy zauważą zmiany `PORTEXEC` i akcje wykorzystujące te zmienne będą dalej wykonywane wewnątrz dawnego miejsca rezydowania

drzewa Portage.

Prekompilowane pakiety

Domyślnie Portage nie korzysta z prekompilowanych pakietów, posiada jednak wsparcie dla nich i istnieje możliwość korzystania z nich wedle potrzeb. Jeśli zażądamy od Portage zbudowania takiej paczki trafi ona do /usr/portage/packages. Ścieżka ta przechowywana jest w zmiennej PKGDIR.

Kod źródłowy

Domyślnie pobrany kod źródłowy instalowanych aplikacji jest przechowywany wewnątrz /usr/portage/distfiles. Tę lokalizację określa zmienna DISTDIR.

Pliki RPM

Portage umożliwia wygenerowanie plików RPM przy pomocy polecenia ebuild (Więcej informacji: [Aplikacja Ebuild](#)). Domyślnym miejscem zapisu wygenerowanych RPM-ów jest /usr/portage/rpm, przechowywany w zmiennej RPMDIR.

Baza Portage

Lista pakietów zainstalowanych w systemie znajduje się w pliku /var/db/pkg. Pod żadnym pozorem nie należy zmieniać ręcznie jego zawartości. Może to poważnie uszkodzić Portage.

Cache Portage

Cache Portage (informacje o zmianach w plikach, virtualach, drzewie zależności itp.) znajduje się w katalogu /var/cache/edb. Katalog ten można wyczyścić tylko wtedy, gdy nie jest uruchomiona żadna związana z pracą z Portage aplikacja.

1.c. Budowanie programów

Tymczasowe pliki Portage

Tymczasowe pliki Portage zapisywane są domyślnie w katalogu /var/tmp, do którego ścieżkę przechowuje zmienna PORTAGE_TMPDIR.

Zmiana PORTAGE_TMPDIR powinna nieść ze sobą zmianę szeregu innych zmiennych, które przechowują ścieżki do katalogów wewnątrz starej lokalizacji katalogu tymczasowego. Spowodowane jest to sposobem zarządzania zmiennymi BUILD_PREFIX przez Portage.

Tworzenie katalogów

Tymczasowe, osobne dla każdego budowanego pakietu katalogi powstają w /var/tmp/portage. Miejsce to zapisane jest w zmiennej BUILD_PREFIX.

Lokalizacja systemu plików

Domyślnie Portage instaluje pakiety w bieżącym systemie plików (/), można jednak to zmienić ustawiając zmienną środowiskową ROOT. Przydaje się to, gdy chcemy stworzyć nowe obrazy budowy.

1.d. Logowanie zdarzeń

Logowanie Ebuild

Portage może tworzyć osobne logi dla każdego ebuildu tylko wtedy, gdy zmienna PORT_LOGDIR wskazuje na katalog, do którego grupa portage (z której prawami uruchamiane są wszystkie procesy) ma prawa zapisu. Domyślnie zmienna ta nie jest ustawiona.

2. Konfigurowanie Portage

2.a. Konfiguracja Portage

Portage konfiguruje się poprzez zmienne, znajdujące się na ogół w pliku /etc/make.conf. Dla uzyskania pełnych informacji nt. tego pliku zalecamy przeczytanie jego man'a:

Listing 1: Wywoływanie man make.conf

```
$ man make.conf
```

2.b. Opcje budowania programów

Opcje komplikacji

W trakcie budowania programu Portage przekazuje kompilatorowi następujące zmienne:

- Zmienna CFLAGS & CXXFLAGS definiują żądane flagi dla komplikacji kodu C i C++
- Zmienna CHOST zawiera informację o hoście na którym budowany jest program
- Zmienna MAKEOPTS jest przekazywana do polecenia make i jej wartość jest najczęściej ilością równoległych zadań podczas komplikacji.

Również zmienne USE są używane podczas budowania programów przez Portage, ale zostały już szczegółowo omówione w poprzednich rozdziałach, więc nie ma potrzeby omawiania ich tutaj po raz kolejny.

Opcje emergowania

Kiedy Portage emerguje nowszą wersję danego programu usuwa przestarzałe pliki z systemu. Usunięcie to jest poprzedzone odpowiednim komunikatem, a użytkownik ma 5 sekund na przerwanie całej operacji i pozostanie przy aktualnej wersji programu. Owe 5 sekund definiowanie jest zmienną CLEAN_DELAY.

2.c. Ochrona plików konfiguracyjnych

Chronione przez Portage katalogi

Jeśli plik nie znajduje się w lokacji chronionej przez Portage to przy instalowaniu nowszej wersji programu, do którego należy zostanie po prostu nadpisany. Te chronione katalogi również możemy skonfigurować, są one przechowywane w zmiennej CONFIG_PROTECT.

Plik znajdujący się w takiej chronionej lokacji nie zostanie nadpisany, Portage zapisze nowy plik pod inną nazwą i poinformuje użytkownika o pojawienniu się nowszej wersji.

Więcej informacji nt. swojego aktualnego ustawienia zmiennej CONFIG_PROTECT dostaniemy po wpisaniu komendy emerge --info:

Listing 2: Znajdowanie aktualnego ustawienie zmiennej CONFIG_PROTECT

```
$ emerge --info | grep 'CONFIG_PROTECT='
```

Więcej informacji na temat ochrony plików konfiguracyjnych w Portage uzyskamy wpisując:

Listing 3: Więcej informacji nt. ochrony plików konfiguracyjnych w Portage

```
$ emerge --help config
```

Odsłanianie chronionych katalogów

Żeby odsłonić konkretny chroniony katalog i umożliwić w nim bezpośrednie nadpisywanie plików dodajmy go do zmiennej CONFIG_PROTECT_MASK.

2.d. Opcje pobierania

Serwery

Jeśli potrzebne są jakieś pliki lub informacje, których nie znajdują się na dysku to Portage będzie zmuszone pobrać je z Internetu. Miejsca, w których program będzie ich szukał definiujemy w następujących zmiennych:

- GENTOO_MIRRORS zawiera adresy serwerów lustrzanych z kodami źródłowymi (distfiles) programów z Portage.
- PORTAGE_BINHOST zawiera adresy serwerów z prekompilowanymi pakietami.

Kolejna zmienna zawiera adres serwera rsync, z którego pobierane będą aktualizacje drzewa Portage:

- Zmienna SYNC zawiera nazwę serwera, z którego Portage będzie pobierało aktualizacje drzewa Portage.

Zmienne GENTOO_MIRRORS i SYNC mogą zostać ustawione przy pomocy programu mirrorselect. Aby móc go używać należy go zainstalować, robimy to poleceniem emerge mirrorselect. Więcej informacji o programie uzyskamy wpisując:

Listing 4: Więcej informacji o mirrorselect

```
# mirrorselect --help
```

Jeśli dodatkowo chcemy korzystać z serwera proxy, używamy do tego zmiennych HTTP_PROXY, FTP_PROXY, RSYNC_PROXY.

Komendy pobierania

Do pobierania kodów źródłowych Portage domyślnie używa programu wget. Możemy to zmienić poprzez zmienną FETCHCOMMAND.

Portage jest w stanie wznowić przerwany transfer. Używa w takim przypadku jednej z możliwości programu wget. Jeśli chcemy to zmienić to wystarczy wyedytować zmienną RESUMECOMMAND.

Należy upewnić się, że wybrane przez nas nowe polecenia FETCHCOMMAND i RESUMECOMMAND umieszczają kody źródłowe w odpowiednich miejscach. Wewnątrz zmiennych powinno się umieścić \${URI} i \${DISTDIR} odpowiednie dla lokacji kodów źródłowych i distfiles.

Można również wybrać osobne komendy pobierania w zależności od protokołu, który akurat jest używany, służą do tego zmienne: FETCHCOMMAND_FTP, RESUMECOMMAND_HTTP, RESUMECOMMAND_FTP, itd.

Ustawienia rsync

Nie można wprawdzie zastąpić innym polecenia rsync używanego do aktualizowania drzewa Portage, ale mamy za to do dyspozycji kilka zmiennych, dzięki którym można dostosować niektóre parametry jego działania.

- RSYNC_EXCLUDEFROM wskazuje konkretne paczki, lub kategorie paczek, które będą ignorowane w procesie aktualniania.
- RSYNC_RETRIES definiuje jak wiele razy rsync będzie próbował wznowić połączenie zanim zrezygnuje, domyślnie jest to 3.
- RSYNC_TIMEOUT definiuje przez ile sekund rsync może pozostawać w stanie bezczynności zanim uzna, że czas oczekiwania minął. Domyślnie jest to 180, ale użytkownicy modemów zapewne zechcą przestawić tą opcję na wartość 300 lub więcej.

2.e. Konfiguracja Gentoo

Wybór gałęzi

Wyboru gałęzi dokonujemy poprzez zmianę zmiennej ACCEPT_KEYWORDS. Domyślnie jest to stabilna gałąź naszej architektury, więcej informacji o innych gałęziach znaleźć można w dalszych rozdziałach podręcznika.

Portage Features

Przy pomocy zmiennej FEATURES aktywujemy rozmaite dodatkowe możliwości Portage, które szerzej są omawiane w poświęconym im rozdziale [Możliwości Portage](#).

2.f. Zachowanie Portage

Zarządzanie zasobami

Zmienna PORTAGE_NICENESS służy do zwiększenia, bądź zmniejszania wartości nice z jaką działa Portage. Wartość ze zmiennej PORTAGE_NICENESS jest dodawana do aktualnej wartości nice.

Więcej informacji o wartościach nice znajdziesz w man'ie programu nice:

Listing 5: Więcej informacji o nice

```
$ man nice
```

Konfiguracja danych wyjściowych

Zmienna NOCOLOR, domyślnie ustawiona na "false" (fałsz), przestawiona na "true" zakaże Portage kolorowania danych wyjściowych.

3. Mieszanie różnych gałęzi Portage

3.a. Gałęzie Portage

Gałiąz stabilna

Zmienna ACCEPT_KEYWORDS definiuje której gałęzi Portage zamierzamy używać w swoim systemie. Domyślnie jest to stabilna gałąź dla naszej architektury, np. x86.

Zwykle, zwłaszcza początkującym użytkownikom zalecamy używanie gałęzi stabilnej. Jeśli jednak nie zależy nam nazbyt na pełnej stabilności i chcemy pomóc deweloperom przesyłając raporty o błędach na stronę <http://bugs.gentoo.org> to zapraszamy do dalszej lektury.

Gałiąz testowa

By używać najświeższego oprogramowania warto zainteresować się testową częścią Portage. Wszystko co należy zrobić, aby przejść na tą wersję to wpisanie ~ (tylko) przed kodem architektury.

Jeżeli deweloper sądzi, że pakiet jest funkcjonalny, ale nie został dokładnie przetestowany to należy on do gałęzi testowej. Każdy odnaleziony w takim pakiecie błąd należy [zgłośić](#) deweloperom Gentoo.

Odmaskowanie całej gałęzi testowej może sprawić, że system stanie się niestabilny, nie wszystkie pakiety będą się prawidłowo instalować (na przykład w związku z brakującymi lub zepsutymi zależnościami), aktualizacje będą musiały odbywać się częściej niż zwykle, a niektóre pakiety po prostu będą zepsute. Rozwiążanie to jest skierowane do bardziej doświadczonych użytkowników.

Na przykład, aby wybrać testową gałąź dla architektury x86 wystarczy wyedytować plik /etc/make.conf i wstawić tam:

Listing 1: Ustawianie zmiennej ACCEPT_KEYWORDS

```
ACCEPT_KEYWORDS=~x86
```

Gdy spróbujemy w tym momencie uaktualnić system, zorientujemy się, że portage chce przeinstalować naprawdę dużo programów. Należy również pamiętać też, że jeśli już uaktualnimy system do wersji testowej nie będzie łatwej drogi powrotnej do oficjalnej wersji stabilnej.

3.b. Mieszanie gałęzi stabilnej i testowej

Plik package.keywords

Można nakazać Portage, aby używało wersji testowych tylko niektórych paczek. Nie ma potrzeby przestawiania całego systemu w ten tryb dla jednego, lub nawet kilku programów. Aby to osiągnąć wystarczy dodać kategorię i nazwę wybranej paczki do pliku /etc/portage/package.keywords. Możemy również utworzyć katalog o takiej samej nazwie i umieścić tam pliki z wpisanyimi do nich pakietami. Na przykład, aby Portage używało wersji niestabilnej programu gnumeric:

Listing 2: /etc/portage/package.keywords - ostateczna wersja ustawienia dla gnumeric

```
app-office/gnumeric ~x86
```

Testowanie określonej wersji programu

Czasem zdarza się tak, że chcemy zainstalować konkretną wersję danego programu, najczęściej z gałęzi niestabilnej i za żadne skarby nie chcemy, aby przy uaktualnieniach Portage instalowało (obojętnie) starszą lub nowszą wersję tego programu. Wtedy bardzo pomocna okazuje się dla nas możliwość wymuszenia na Portage używania tej właśnie wersji, a robimy to poprzez dodanie znaku = na początek linii danego programu w pliku package.keywords. Możemy też używać innych znaków matematycznych dla określenia przedziału, do którego należą żądane przez nas wersje - są to operatory <=, <, > i >=.

W każdym przypadku gdy chcemy dodać informację o wersji trzeba użyć odpowiedniego operatora. Jeśli nie chcemy dodawać żadnych informacji o pożądanych wersjach po prostu nie dodajmy żadnych operatorów.

Na przykład nakażmy portage używać wyłącznie gnumeric-1.2.13:

Listing 3: Włączanie konkretnej wersji testowej gnumeric

```
=app-office/gnumeric-1.2.13
```

3.c. Instalacja zamaskowanych programów

Plik package.unmask

Deweloperzy Gentoo **nie** będą pomagać przy problemach w korzystaniu z tego pliku. Prosimy o zachowanie ostrożności przy wprowadzaniu tych zmian. Nikt nie będzie odpowiadał na pytania związane z plikami package.unmask i package.mask

Jeśli zechcemy zainstalować program z jakichś względów zamaskowany przez developerów Gentoo powinniśmy najpierw zapoznać się z powodem ukrycia danej jego wersji, znajdującym się domyślnie w pliku /usr/portage/profiles, a następnie dodać dokładnie taką samą linię do pliku /etc/portage/package.unmask (lub pliku w tym katalogu, jeśli jest on katalogiem o takiej nazwie).

Na przykład jeśli =net-mail/hotwayd-0.8 jest zamaskowane można je odmaskować dodając taką linię do lokalizacji package.unmask:

Listing 4: /etc/portage/package.unmask

```
=net-mail/hotwayd-0.8
```

Plik package.mask

Jeśli z jakichś powodów nie życzymy sobie, aby Portage aktualniało program do jakiejś określonej wersji możemy ją zamaskować dopisując odpowiednią linię do pliku /etc/portage/package.mask (lub w tym pliku w tym katalogu).

Na przykład jeśli nie chcemy, aby Portage instalowało nowsze źródła kernela niż gentoo-sources-2.6.8.1 dodajemy taką linię do lokalizacji package.mask:

Listing 5: Przykładowy wpis do /etc/portage/package.mask

```
>sys-kernel/gentoo-sources-2.6.8.1
```

4. Dodatkowe narzędzia Portage

4.a. Etc-update

Etc-update jest narzędziem, które służy do zastępowania plików konfiguracyjnych plikami `_.cfg0000_<nazwa>`, umożliwia ich interaktywną edycję, oraz pozwala automatycznie dokonać drobnych zmian w owych plikach. Pliki `_.cfg0000_<nazwa>` są generowane przez Portage, gdy chce nadpisać jakiś plik w katalogu chronionym zmienną CONFIG_PROTECT.

Aby uruchomić etc-update wystarczy po prostu wpisać:

Listing 1: Uruchamianie etc-update

```
# etc-update
```

Program automatycznie dokona drobnych zmian w plikach konfiguracyjnych, a potem pokaże listę plików chronionych i poprosi o decyzję w ich sprawie. Na dole pojawi się poniższa lista dostępnych opcji wraz z ich krótkim opisem:

Listing 2: Opcje etc-update

```
Please select a file to
edit by entering the corresponding number.
(-1 to exit)
(-3 to auto merge all remaining files)
(-5 to auto-merge AND not use 'mv -i'):
```

Po wybraniu -1 etc-update zakończy działanie. Warto pamiętać, że jest to jedynie polecenie zakończenia programu i nie cofnie żadnych dokonanych wcześniej zmian. Po wybraniu -3 lub -5 wszystkie znajdujące się na liście pliki konfiguracyjne zostaną nadpisane nowszymi wersjami. Dobrym pomysłem jest zaznaczenie plików, których nie chcemy nadpisywać automatycznie. Dokonuje się tego po prostu wpisując liczbę znajdująca się na lewo od danego pliku.

Np. wybieramy sobie plik konfiguracyjny /etc/pear.conf i po wybraniu jego indeksu widzimy coś takiego:

Listing 3: Oddzielne uaktualnienie wybranego pliku

```
Beginning of differences between /etc/pear.conf and /etc/_.cfg0000_pear.conf
[...]
End of differences between /etc/pear.conf and /etc/_.cfg0000_pear.conf
1) Replace original with update
2) Delete update, keeping original as is
3) Interactively merge original with update
4) Show differences again
```

W ten sposób można łatwo uzyskać informacje o różnicach pomiędzy oboma plikami. Jeśli jesteśmy pewni, że zastąpienie starego pliku nowym to dobry pomysł, naciskamy 1. Może zdarzyć się też tak, że nie będziemy chcieli nowego pliku. Wtedy naciskamy 2 i zapominamy o tym, że była nowsza wersja :) Jeśli chcemy bliżej zająć się tym plikiem (tzw. metoda interaktywna) wybieramy 3.

Nie ma sensu rozpisywać się na temat trzeciej metody - ograniczymy się jedynie do podania możliwych w tym trybie do wybrania komend. Generalnie wygląda to tak, że program pokazuje dwie linie - oryginalną i proponowaną i czeka aż wpiszemy jeden z ciągów znaków:

Listing 4: Komendy dostępne podczas interaktywnej edycji plików

ed:	Edycja i użycie obu wersji, każdej z nagłówkiem.
eb:	Edycja i użycie obu wersji.
el:	Edycja i użycie wersji po lewej.
er:	Edycja i użycie wersji po prawej.
e:	Edycja nowej wersji.
l:	Użycie wersji po lewej.
r:	Użycie wersji po prawej.
s:	Dołączenie wspólnych linii bez informowania o tym.
v:	Dołączenie wspólnych linii z podaniem informacji.
q:	Zakończenie.

Kiedy już skończymy uaktualniać te najważniejsze pliki, pozostałe możemy zamienić w trybie automatycznym. Etc-update wyłączy się kiedy już nie będzie miało żadnych plików do uaktualnienia.

4.b. Dispatch-conf

Dispatch-conf to ciekawa alternatywa dla etc-update. Program posiada wszystkie opcje swojego pierwotnego, a ponadto jest w stanie zapamiętać kolejne zmiany jakich dokonaliśmy w naszych plikach konfiguracyjnych, dzięki czemu w razie jakiejś katastrofy możemy łatwo wrócić do poprzedniej wersji.

Tak jak w etc-update, można zażądać, aby program pozostawił dany plik bez zmian, zastąpić go nowym, edytować ręcznie obecny plik lub użyć do tego celu trybu interaktywnego. Oprócz tego dispatch-conf ma jeszcze kilka ciekawych opcji:

- Automatycznie zamienia stare pliki nowymi jeśli zmiany w nich dotyczą jedynie linii wykomentowanych.
- Automatycznie zamienia pliki, gdy zmiany dotyczą jedynie pustego miejsca (spacje, tabulatory, puste wiersze itp.)

Pracę z programem należy zacząć od wprowadzenia odpowiednich ustawień do pliku /etc/dispatch-conf.conf, w tym przede wszystkim zmiennej archive-dir, wskazującej katalog gdzie program będzie archiwizował nadpisywane pliki konfiguracyjne oraz utworzenia owego katalogu.

Więcej informacji o programie dostarczy man dispatch-conf:

Listing 5: Czytanie manuala dispatch-conf

```
$ man dispatch-conf
```

4.c. Quickpkg

Program quickpkg umożliwia spakowanie zainstalowanego programu do paczki, z której następnie możemy go bezproblemowo i błyskawicznie odtworzyć. Uruchamianie quickpkg jest proste: po prostu podajemy nazwy programów do spakowania jako parametry i wciskamy enter.

Na przykład wybieramy do spakowania: curl, arts i procps:

Listing 6: Przykład użycia quickpkg

```
# quickpkg curl arts procps
```

Po zakończeniu całego procesu gotowe paczki znajdziemy w katalogu \$PKGDIR/All (domyślnie /usr/portage/packages/All). Ponadto dla naszej wygody w \$PKGDIR/<kategoria> będą się znajdowały dowiązania symboliczne do wszystkich zbudowanych przez nas paczek.

5. Pozostawiając oficjalne drzewo Portage

5.a. Używanie podzestawów drzewa Portage

Pomijanie kategorii/pakietów

Możemy też selektywnie aktualnić poszczególne kategorie/pakiety oraz zignorować pozostałe kategorie/pakiety. Osiągamy to zmuszając rsync do pominięcia kategorii/pakietów podczas wykonywania emerge --sync.

W pliku /etc/make.conf można skonfigurować zmienną RSYNC_EXCLUDEFROM, która powinna zawierać ścieżkę do pliku, w którym znajdują się informacje o kategoriach i pakietach, które mają być pomijane przy aktualizowaniu drzewa.

Listing 1: Definiowanie pliku z pominiętymi pakietami w make.conf

```
RSYNC_EXCLUDEFROM=/etc/portage/rsync_excludes
```

Listing 2: Wyłączanie wszystkich gier w pliku /etc/portage/rsync_excludes

```
games-*/*
```

Należy zwrócić uwagę, że może to doprowadzić do problemów z zależnościami, gdyż nowe, niepominięte pakiety mogą zależeć od nowych lecz pominiętych pakietów.

5.b. Dodawanie nieoficjalnych ebuildów

Definiowanie katalogu-nakładki na Portage

Można zmusić Portage, aby używało ebuildów, które nie są dostępne w oficjalnym drzewie. Należy utworzyć nowy katalog (na przykład /usr/local/portage) w którym będą znajdować się dodatkowe ebuildy. Należy pamiętać o zachowaniu struktury katalogów takiej jak w oficjalnym drzewie Portage.

Następnie trzeba zdefiniować zmienną PORTDIR_OVERLAY w pliku /etc/make.conf tak żeby wskazywała na właściwie utworzony katalog. Możliwe jest teraz użycie tych ebuildów bez obawy, że zostaną usunięte lub nadpisane przy następnym uruchomieniu emerge --sync.

Praca z kilkoma powłokami

Dla zaawansowanych użytkowników będących deweloperami i korzystających z kilku powłok, testujących pakiety przed dodaniem ich do Portage lub chcących po prostu używać nieoficjalnych ebuildów pochodzących z różnych źródeł, z pakietem app-portage/gentoolkit-dev zostaje dostarczony gensync, narzędzie, dzięki któremu będziemy mogli utrzymać nasze repozytorium aktualnym.

gensync daje nam możliwość aktualizacji wszystkich repozytoriów za jednym razem lub wybranie tylko kilku z nich. Każde repozytorium powinno posiadać plik .syncsource w katalogu konfiguracyjnym /etc/gensync/, w którym znajduje się lokalizacja repozytorium, nazwa, ID, itp.

Przypuśćmy, że posiadamy dwa repozytoria nazwane java (dla ebuildów java) oraz entapps (dla aplikacji rozwijanych w warunkach domowych, jednak na potrzeby przedsiębiorstw). Aktualizację, możemy przeprowadzić w następujący sposób:

Listing 3: Użycie gensync do aktualizacji repozytoriów

```
# gensync java entapps
```

5.c. Programy spoza Portage

Portage kontra własnoręcznie zarządzane pakiety

W niektórych przypadkach chcemy skonfigurować, zainstalować i zarządzać programami samodzielnie, bez pomocy Portage w automatyzacji tego procesu, nawet jeśli Portage zawiera te programy. Najczęściej są to źródła jądra i sterowniki nvidii. Można skonfigurować Portage tak by myślało, że dany pakiet jest zainstalowany w systemie. Ten proces nazywany jest wstrzykiwaniem i jest obsługiwany przez Portage dzięki plikowi /etc/portage/profile/package.provided.

Dla przykładu, jeśli chcemy poinformować Portage, że ręcznie zainstalowaliśmy development-sources-2.6.11.6 dodajemy następującą linijkę do /etc/portage/profile/package.provided:

Listing 4: Przykładowa linijka dla pliku package.provided

```
sys-kernel/development-sources-2.6.11.6
```

D. Konfiguracja sieci w Gentoo

1. Wprowadzenie

1.a. Początek

Uwaga: Ten dokument zakłada, że jądro zostało poprawnie skonfigurowane, że prawidłowo zainstalowano również jego moduły dla sprzętu oraz, że znana jest nazwa interfejsu sprzętowego. Zakłada się również, że konfigurowane jest eth0, ale może to być również eth1, wlan0, etc.

Uwaga: Przy pisaniu dokumentu zakładamy, że jest zainstalowany baselayout-1.11.11 lub nowszy.

Przed rozpoczęciem konfiguracji karty sieciowej, należy wspomnieć o systemowym RC w Gentoo. Jest to realizowane poprzez stworzenie linku symbolicznego z net.lo do net.eth0 w /etc/init.d.

Listing 1: Tworzenie połączenia symbolicznego między net.eth0 i net.lo

```
# cd /etc/init.d  
# ln -s net.lo net.eth0
```

System RC w Gentoo już wie o tym interfejsie. Musi również wiedzieć jak skonfigurować nowy interfejs. Wszystkie interfejsy sieciowe są konfigurowane w /etc/conf.d/net. Poniżej znajduje się przykładowa konfiguracja dla DHCP oraz statycznych adresów.

Listing 2: Przykłady dla /etc/conf.d/net

```
# Dla DHCP  
config_eth0=( "dhcp" )  
  
# Dla statycznego IP używając notacji CIDR  
config_eth0=( "192.168.0.7/24" )  
routes_eth0=( "default via 192.168.0.1" )  
  
# Dla statycznego IP używając notacji netmaski  
config_eth0=( "192.168.0.7 netmask 255.255.255.0" )  
routes_eth0=( "default gw 192.168.0.1" )
```

Uwaga: Jeżeli nie zostanie określona konfiguracja dla interfejsu, zakłada się że użyte zostanie DHCP.

Uwaga: CIDR oznacza Classless InterDomain Routing. Początkowo adresy IPv4 były podzielone na klasy A, B lub C. Klasyfikacja ta nie przewidywała takiego wzrostu popularności Internetu i teraz stoi przed obliczem utraty unikalnych adresów IP. CIDR pozwala użyć jednego schematu adresowania w celu użycia jednego adresu IP do reprezentowania wielu adresów. Adres IP w notacji CIDR wygląda jak zwykły adres IP, z tą różnicą, że kończy się ukośnikiem za którym znajduje się liczba. Dla przykładu, 192.168.0.0/16. CIDR jest dokładnie opisany w [RFC 1519](#).

Teraz, gdy interfejs został już skonfigurowany, można go uruchomić i zatrzymać używając poleceń znajdujących się poniżej.

Listing 3: Uruchamianie i zatrzymywanie sieci przy pomocy skryptów startowych

```
# /etc/init.d/net.eth0 start  
# /etc/init.d/net.eth0 stop
```

Ważne: W przypadku problemów z siecią, zaleca się ustawienie zmiennej **RC_VERBOSE="yes"** w /etc/conf.d/rc, aby uzyskać więcej informacji na temat tego, co się dzieje.

Teraz, gdy już udało się uruchomić oraz zatrzymać urządzenie sieciowe, należałoby dodać je do domyślnych skryptów startowych Gentoo. Poniżej opisane jest jak tego dokonać. Ostatnie polecenie "rc" powoduje uruchomienie tych skryptów w danym poziomie uruchomieniowym, które jeszcze nie zostały jeszcze uruchomione.

Listing 4: Konfiguracja interfejsu sieciowego, aby uruchamiał się przy starcie systemu

```
# rc-update add net.eth0 default
# rc
```

2. Zaawansowana konfiguracja

2.a. Zaawansowana konfiguracja

Zmienna config_eth0 jest sercem konfiguracji interfejsu sieciowego. Jest to lista poleceń konfiguracyjnych wysokiego poziomu (w tym przypadku urządzenia eth0). Każde polecenie z listy poleceń jest uruchamiane w sposób sekwencyjny. Urządzenie uruchomi się jeżeli co najmniej jedno polecenie zostanie poprawnie uruchomione.

Poniżej znajduje się lista wbudowanych poleceń.

Polecenie	Opis
null	Nie robi nic
noop	Jeżeli urządzenie działa i jest przypisany adres, zakończy pomyślnie konfigurację.
adres IPv4 lub IPv6	Dodaje wskazany adres do interfejsu
dhcp, adsl lub apipa (lub dowolne polecenie pochodzące z modułu producenta)	Uruchamia moduł, który posiada dane polecenie. Dla przykładu, dhcp uruchomi moduł, który zapewnia DHCP i który może być którymś z grupy dhcpcd, udhcpc, dhclient lub pump.

Jeżeli jakieś polecenie się nie wykona, można zdefiniować takie które będzie wykonywane zamiennie. Polecenie to musi pasować dokładnie do struktury konfiguracji głównej.

Można połączyć te polecenia razem. Poniżej znajduje się kilka przykładów.

Listing 1: Przykłady konfiguracji

```
# Dodawanie trzech adresów IPv4
config_eth0=(
    "192.168.0.2/24"
    "192.168.0.3/24"
    "192.168.0.4/24"
)
# Dodawanie adresu IPv4 oraz dwóch adresów IPv6
config_eth0=(
    "192.168.0.2/24"
    "4321:0:1:2:3:4:567:89ab"
    "4321:0:1:2:3:4:567:89ac"
)

# Zachowuje przypisany adres, chyba że urządzenie zostanie wyłączone
# - w takim wypadku należy przypisać kolejny adres poprzez DHCP.
# Jeżeli pobranie adresu przez DHCP nie powiedzie się - zostanie
# przypisany stały adres IP poprzez APIPA
config_eth0=(
    "noop"
    "dhcpc"
)
fallback_eth0=(
    "null"
    "apipa"
)
```

Uwaga: Przy używaniu modułu ifconfig oraz dodawaniu więcej niż jednego adresu zostają utworzone aliasy dla każdego dodatkowego adresu. Wobec tego, powyższe dwa przykłady utworzą interfejsy eth0, eth0:1 oraz eth0:2. Nie można nic specjalnego z tymi interfejsami zrobić, gdyż jądro oraz programy będą traktować interfejsy eth0:1 oraz eth0:2 jako eth0.

Ważne: Kolejność zapasowej konfiguracji jest bardzo ważna! Gdyby polecenie null nie zostało zdefiniowane, to polecenie apipa zostały wykonane tylko w przypadku gdyby polecenie noop nie powiodło się.

Uwaga: APIPA oraz DHCP będą omawiane później.

2.b. Zależności sieciowe

Skrypty startowe znajdujące się w /etc/init.d mogą być zależne od konkretnego urządzenia sieciowego lub po prostu od usługi net. Usługa net może być zdefiniowana w /etc/conf.d/rc za pomocą zmiennej RC_NET_STRICT_CHECKING i może oznaczać różne rzeczy.

Wartość Opis

none	Zakłada, że sieć jest zawsze włączona
no	Oznacza, że co najmniej jedna usługa sieciowa oprócz net.lo musi być włączona. Opcja ta może być używana przez właścicieli komputerów przenośnych z kartami wifi oraz zwykłymi kartami sieciowymi, w których powinno być uruchomione jednocześnie tylko jedno urządzenie.
lo	Działa podobnie jak opcja no, z tą różnicą, że net.lo również jest wliczane. Jest to szczególnie przydatne dla osób, którym nie robi różnicy czy uruchamia się jakiekolwiek urządzenie sieciowe.
yes	Ta opcja oznacza, że WSZYSTKIE urządzenia sieciowe MUSZĄ być uruchomione, aby można było uznać usługę net za działającą.

Ale co z net.br0 zależnym od net.eth0 oraz net.eth1? net.eth1 może być urządzeniem bezprzewodowym lub ppp, które potrzebuje skonfigurowania zanim zostanie uruchomione. Czynność ta nie może być dokonana w /etc/init.d/net.br0, gdyż jest to link symboliczny do net.lo.

Rozwiązaniem tego problemu jest samodzielne stworzenie funkcji depend() w /etc/conf.d/net

Listing 2: Zależność net.br0 w /etc/conf.d/net

```
# Można użyć dowolnej zależności (use, after, before) według przykładów znalezionych w skryptach
depend_br0() {
    need net.eth0 net.eth1
}
```

Więcej informacji o zależnościach można znaleźć w sekcji dotyczącej tworzenia skryptów inicjacyjnych w Podręczniku Gentoo.

2.c. Nazwy zmiennych i ich wartości

Nazwy zmiennych są dynamiczne. Najczęściej posiadają one strukturę zmienna_\${interfejs|mac|essid|apmac}. Przykładowo, zmienna dhcpcd_eth0 przechowuje wartość dla opcji dhcpcd dla interfejsu eth0, zaś dhcpcd_essid przechowuje wartości dla opcji dhcpcd gdy interfejs podłączy się do ESSID o nazwie "essid".

Jednakże, nie ma zasad mówiącej o tym, iż nazwy interfejsów muszą mieć format ethx. Wiele urządzeń bezprzewodowych posiadają nazwy takie jak wlanx, rax, jak również eth.x. Dodatkowo, niektóre interfejsy sieciowe zdefiniowane przez użytkowników, takie jak mostki, mogą posiadać dowolną nazwę, np. foo. Aby urozmaicić życie, bezprzewodowe punkty dostępu mogą mieć nazwy ze znakami nie alfanumerycznymi - jest to ważne, gdyż część opcji można konfigurować dla konkretnego ESSID-a.

Na domiar złego, Gentoo używa zmiennych bashowych do kontrolowania sieci - a bash nie potrafi korzystać z niczego co pochodzi spoza angielskich znaków alfanumerycznych. Aby ominąć te ograniczenie, każdy znak pochodzący spoza znaków dopuszczalnych zamieniany jest na znak _.

Kolejnym ograniczeniem powłoki bash jest to, że niektóre ze znaków muszą być specjalnie cytowane, czyli musi pojawić się przed nimi symbol \. Znaki, których to dotyczy to ", ' oraz \.

W poniższym przykładzie, zostaje użyty bezprzewodowy ESSID z najszerzym możliwym zestawem znaków. Zostanie użyty ESSID My "\ NET:

Listing 3: przykład nazewnictwa zmiennej

```
# Poniższe działa, ale domena jest nieprawidłowa
dns_domain_My____NET="My \"\\ NET"

# Powyższe ustawienia ustawiają domenę dns jako My "\ NET gdy karta
#bezprzewodowa połączy się z punktem dostępu którego ESSID to My "\ 
#NET
```

3. Modularna praca w sieci

3.a. Moduły sieciowe

Obecnie wspierane są modułowe skrypty sieciowe, co oznacza, że w prosty sposób można dodawać kolejne urządzenia sieciowe i moduły konfiguracyjne, zachowując zgodność z obecnie działającymi.

Moduły są domyślnie wczytywane w momencie gdy są potrzebne przez jakiś pakiet. Jeżeli zostanie zdefiniowany moduł, który nie posiada zainstalowanego pakietu, wyświetlony zostanie błąd z komunikatem mówiącym jaki pakiet należy doinstalować. Najczęściej ustawień modułów używa się jedynie wtedy, gdy zostały zainstalowane dwa lub więcej pakiety, które udostępniają tę samą usługę i należy wyznaczyć która usługa ma pierwszeństwo.

Uwaga: Wszystkie omówione w tym rozdziale ustawienia powinny być wpisane do pliku /etc/conf.d/net, chyba, że zaznaczymy inaczej.

Listing 1: Ustawienia modułów

```
# iproute2 ważniejsze niż ifconfig
modules=( "iproute2" )

# Można również określić inne moduły dla interfejsu.
# W tym przypadku, udhcpc jest ważniejsze niż dhcpcd
modules_eth0=( "udhcpc" )

# Możliwe jest również określenie których modułów nie używać wcale -
# przykładowo może być używana kliencka lub kontrolowana przez linux-wlan-ng
# konfiguracja wifi, jednakże zachodzi potrzeba skonfigurowania własnych
# ustawień sieciowych dla każdego ESSID-a z którym sieć jest powiązana osobno.

modules=( "!iwconfig" )
```

3.b. Kontrolery sieciowe

Dostępne są dwa pakiety służące do kontrolowania interfejsów sieciowych: ifconfig oraz iproute2. Potrzebne jest jedno z tych dwóch, aby cokolwiek skonfigurować na urządzeniu sieciowym.

Domyślnie w Gentoo używane jest ifconfig i jest dostępny w profilu systemowym. iproute2 jest potężniejszy i elastyczniejszy, ale nie jest załączany domyślnie.

Listing 2: Aby zainstalować iproute2

```
# emerge sys-apps/iproute2

# Aby iproute2 miało wyższy priorytet niż ifconfig, w przypadku gdy
# obydwa są zainstalowane
modules=( "iproute2" )
```

Jako że ifconfig oraz iproute2 są podobne w działaniu, można pozwolić, aby ich podstawowa konfiguracja współpracowała ze sobą. Dla przykładu, poniższe linijki współpracują z obydwoma programami.

Listing 3: Przykłady dla ifconfig oraz iproute2

```
config_eth0=( "192.168.0.2/24" )
config_eth0=( "192.168.0.2 netmask 255.255.255.0" )

# Można również zdefiniować adres broadcast
config_eth0=( "192.168.0.2/24 brd 192.168.0.255" )
config_eth0=( "192.168.0.2 netmask 255.255.255.0 broadcast 192.168.0.255" )
```

3.c. DHCP

DHCP to pobieranie informacji o sieci (adres IP, serwery DNS, bramka, etc.). Oznacza to, że jeżeli jest serwer DHCP w sieci, należy wskazać komputerom klienckim, aby używały serwera DHCP, dzięki czemu sieć zostanie skonfigurowana automatycznie. Oczywiście, samodzielnie trzeba będzie skonfigurować takie rzeczy jak sieć bezprzewodowa, PPP czy inne, które są wymagane, zanim będzie można skorzystać z DHCP.

Z DHCP można skorzystać za pomocą dhcpcd, dhclient, pump lub udhcpc. Każdy z nich posiada swoje zalety i wady. Oto krótkie wprowadzenie.

Moduł DHCP	Pakiet	Zalety	Wady
dhclient	net-misc/dhc p	Stworzone przez ISC, te same osoby które stworzyły BIND DNS Bardzo konfigurowalne.	Konfiguracja jest bardzo skomplikowana, oprogramowanie jest dość duże, nie można otrzymać serwerów NTP z DHCP i domyślnie nie jest wysyłana nazwa hosta.
dhcpcd	net-misc/dhc pcd	Przez długi czas jako domyślny w Gentoo, nie związany z zewnętrznymi narzędziami	Brak wsparcia ze strony twórców, bywa że jest powolne, nie zawsze potrafi przechodzić w tryb demona.
pump	net-misc/pu mp	Niewielkie rozmiary, nie związany z zewnętrznymi narzędziami	Brak wsparcia ze strony twórców, źle działa przy połączeniach modemowych, nie można otrzymać serwerów NIS z DHCP
udhcpc	net-misc/udh cp	Niewielkie rozmiary - najmniejszy dostępny klient dhcpcd, stworzony dla systemów wbudowanych	Żadna dystrybucja nie używa go jako domyślnego, nie można ustawić czasu wygaśnięcia dłuższego niż 3 sekundy

Jeżeli jest zainstalowanych więcej niż jeden klient DHCP, należy określić który ma być używany. W innym przypadku zostanie użyty dhcpcd, jeżeli jest dostępny.

W celu wysłania określonych opcji do modułu dhcp, należy użyć module_eth0="..." (należy zmienić module na nazwę modułu dhcp, który jest używany, np. dhcpcd_eth0).

Dokładamy starań, aby DHCP było możliwe agnostyczne - wspieramy wobec tego następujące polecenia używając zmiennej dhcp_eth0. Domyślnie żadna z tych zmiennych nie jest ustawiona.

- release - uwalnia adres IP do ponownego użytku
- nodns - nie nadpisuje /etc/resolv.conf
- nontp - nie nadpisuje /etc/ntp.conf
- nonis - nie nadpisuje /etc/yp.conf

Listing 4: Przykładowa konfiguracja DHCP w /etc/conf.d/net

```
# Potrzebne tylko wtedy, gdy jest więcej niż jeden moduł DHCP
modules=( "dhcpcd" )

config_eth0=( "dhcp" )
dhcpcd_eth0="-t 10" # Wygaśnięcie po 10 sekundach
dhcp_eth0="release nodns nontp nonis" # Zdobywa jedynie adres
```

Uwaga: dhcpcd, udhcpc oraz pump domyślnie wysyłają aktualną nazwę hosta do serwera DHCP wobec którego nie trzeba tego definiować.

3.d. Modem ADSL

Na początek należy zainstalować oprogramowanie do ADSL-a.

Listing 5: Instalacja paczki rp-pppoe

```
# emerge net-dialup/rp-pppoe
```

Ostrzeżenie: baselayout-1.11.x wspiera jedynie PPPOE Być może w przyszłości będzie również wsparcie dla PPPOA.

Teraz należy skonfigurować eth0, aby było interfejsem ADSL oraz wstawić nazwę użytkownika. Dokonamy tego edytując plik /etc/conf.d/net.

Listing 6: Konfiguracja eth0 dla ADSL w /etc/conf.d/net

```
config_eth0=( "adsl" )
adsl_user_eth0="użytkownik"
```

Na końcu należy ustawić nazwę użytkownika oraz hasło w /etc/ppp/pap-secrets

Listing 7: sample /etc/ppp/pap-secrets

```
# Znak * jest ważny
"użytkownik" * "hasło"
```

3.e. APIPA {Automatyczne prywatne adresowanie IP (ang. Automatic Private IP Addressing)}

APIPA stara się znaleźć wolny adres w zakresie 169.254.0.0-169.254.255.255 poprzez losowe odpytywanie sieci za pomocą danego interfejsu. Jeżeli nie ma żadnej odpowiedzi, taki adres jest przypisywany do interfejsu.

Przydaje się tylko w sieciach LAN gdzie nie ma serwera DHCP, które nie mają połączenia z Internetem i gdzie wszystkie komputery używają APIPA.

Aby było wsparcie dla APIPA, należy zainstalować net-misc/iptables lub net-analyzer/arping.

Listing 8: Konfiguracja APIPA w /etc/conf.d/net

```
# Najpierw próbujemy skonfigurować poprzez DHCP – jeżeli to się nie powiedzie, próbujemy APIPA
config_eth0=( "dhcp" )
fallback_eth0=( "apipa" )

# Należy użyć jedynie APIPA
config_eth0=( "apipa" )
```

3.f. Wiązanie urządzeń sieciowych

Aby mieć możliwość łączenia urządzeń sieciowych, należy zainstalować net-misc/ifenslave.

Łączenie urządzeń sieciowych stosuje się w celu zwiększenia przepustowości sieci. Jeżeli w komputerze są do dyspozycji dwie karty sieciowe znajdujące się w tej samej sieci, można je połączyć tak, żeby aplikacje w rzeczywistości używały obu urządzeń jednocześnie.

Listing 9: Konfiguracja łączenia w /etc/conf.d/net

```
Aby połączyć urządzenia razem
slaves_bond0="eth0 eth1 eth2"

# Jeżeli nie trzeba przypisywać adresu IP do interfejsu
config_bond0=( "null" )

# Zależne od eth0, eth1 oraz eth2 jako, że mogą wymagać dodatkowej konfiguracji
depend_bond0() {
    need net.eth0 net.eth1 net.eth2
}
```

3.g. Mostkowanie (wsparcie dla 802.1d)

Aby mieć możliwość mostkowania, należy zainstalować net-misc/bridge-utils.

Mostkowanie jest używane do łączenia w całość dużych sieci. Dla przykładu można mieć serwer, który łączy się z internetem przy pomocy ADSL oraz ma połączenie z bezprzewodową kartą sieciową by umożliwić innym komputerom łączenie się z internetem przy pomocy modemu ADSL. Można stworzyć mostek do połączenia obydwu interfejsów.

Listing 10: Konfiguracja mostka w /etc/conf.d/net

```
# Konfiguracja mostka - "man brctl", aby uzyskać szczegółowe informacje
brctl_br0=( "setfd 0" "sethello 0" "stp off" )

# Aby dodać porty do mostka br0
bridge_br0="eth0 eth1"

# Należy skonfigurować porty jako wartość null tak aby dhcp nie uruchomiło się
config_eth0=( "null" )
config_eth1=( "null" )

# Na koniec należy nadać mostkowi adres - można również użyć DHCP
config_br0=( "192.168.0.1/24" )

# Należy zależeć od eth0 oraz eth1 jako że mogą wymagać dodatkowej konfiguracji
depend_br0() {
    need net.eth0 net.eth1
}
```

Ważne: Aby korzystać z niektórych ustawień mostków, warto zjrzeć do dokumentacji opisującej nazwy zmiennych.

3.h. Adresy MAC

W celu zmiany adresów MAC interfejsów sieciowych wystarczy posiadać zainstalowany sys-apps/baselayout-1.11.14 lub nowszy. Jeżeli zachodzi potrzeba zamiany adresu na losowy lub baselayout jest starszy od wyżej wymienionej wersji, należy zainstalować net-analyzer/macchanger.

Listing 11: Przykład zmiany adresu MAC

```
# Aby przypisać adres MAC konkretnemu urządzeniu
mac_eth0="00:11:22:33:44:55"

# Aby tylko ostatnie trzy bajty były losowe
mac_eth0="random-ending"

# Aby wybierać losowo pomiędzy tym samym fizycznym połączeniem (np.
# światłowód, miedź lub bezprzewodowo), wszyscy producenci
mac_eth0="random-samekind"

# Aby wybierać losowo pomiędzy różnymi fizycznymi połączeniami (np.
# światłowód, miedź lub bezprzewodowo), wszyscy producenci
mac_eth0="random-anykind"

# Pełna losowość - UWAGA: niektóre adresy MAC wygenerowane w ten sposób
# mogą zachowywać się inaczej niż powinny
mac_eth0="random-full"
```

3.i. Tunelowanie

Nie trzeba niczego instalować, aby korzystać z tunelowania, gdyż kontroler sieciowy posiada już tę możliwość.

Listing 12: Konfiguracja tunelowania w /etc/conf.d/net

```
# Dla tuneli GRE
iptunnel_vpn0="mode gre remote 207.170.82.1 key 0xffffffff ttl 255"

# Dla tuneli IPIP
iptunnel_vpn0="mode ipip remote 207.170.82.2 ttl 255"

# Aby skonfigurować interfejs
config_vpn0=( "192.168.0.2 peer 192.168.1.1" )
```

3.j. VLAN (wsparcie dla 802.1q)

Aby posiadać wsparcie dla VLAN, należy zainstalować net-misc/vconfig.

VLAN to grupa urządzeń sieciowych które zachowują się tak, jakby były podłączone do jednego segmentu sieciowego - nawet jeśli tak nie jest. Członkowie VLAN-u mogą jedynie widzieć innych członków VLAN-u, nawet jeśli współdzielą sieć z innymi urządzeniami.

Listing 13: Konfiguracja VLAN w /etc/conf.d/net

```
# Wyznaczanie numerów VLAN dla urządzeń
# Należy być pewnym, że ID VLAN-u NIE składają się z zera
vlans_eth0="1 2"

# Można również skonfigurować sam VLAN
# wystarczy zajrzeć do manuala vconfig, aby uzyskać szczegółowe
# informacje
vconfig_eth0=( "set_name_type VLAN_PLUS_VID_NO_PAD" )
vconfig_vlan1=( "set_flag 1" "set_egress_map 2 6" )

# Konfiguracja interfejsu w tradycyjny sposób
config_vlan1=( "172.16.3.1 netmask 255.255.254.0" )
config_vlan2=( "172.16.2.1 netmask 255.255.254.0" )
```

Ważne: Aby używać niektórych ustawień VLAN, może zajść potrzeba zjrzenia do dokumentacji opisującej nazwy zmiennych.

4. Połączenia bezprzewodowe

4.a. Wstęp

Obecnie wspierane są ustawienia dla sieci bezprzewodowych poprzez wireless-tools lub wpa_supplicant. Ważne jest, aby pamiętać że urządzenia bezprzewodowe są konfigurowane globalnie, a nie dla konkretnych urządzeń.

wpa_supplicant jest najlepszym wyborem, jednakże nie wspiera wszystkich sterowników. Pod adresem [projektu wpa_supplicant](#) można uzyskać listę zgodnych urządzeń. Dodatkowo, wpa_supplicant może łączyć się jedynie z ESSID-ami dla których został skonfigurowany.

wireless-tools wspiera niemalże wszystkie karty oraz sterowniki, ale nie potrafi połączyć się z access pointami, które korzystają tylko z WPA.

Ostrzeżenie: Sterownik linux-wlan-ng nie jest w chwili obecnej wspierany przez baselayout. Spowodowane to jest tym, że linux-wlan-ng posiada zestaw własnych ustawień i konfigurację która jest zupełnie inna od pozostałych. Developerzy linux-wlan-ng są namawiani do przejścia na ustawienia zgodne z wireless-tools - gdy to zostanie dokonane, baselayout pozwoli na użycie linux-wlan-ng.

4.b. WPA Supplicant

[WPA Supplicant](#) to pakiet, który pozwala połączyć się do punktów dostępowych z włączonym WPA. Jego ustawienia są dość płynne, ponieważ jest jeszcze w fazie beta - mimo to, działa całkiem dobrze.

Listing 1: Instalacja wpa_supplicant

```
# emerge net-wireless/wpa_supplicant
```

Ważne: Należy mieć włączoną opcję CONFIG_PACKET w jądrze, aby wpa_supplicant działało.

Teraz należy skonfigurować /etc/conf.d/net i wskazać, że wpa_supplicant ma być używany w pierwszej kolejności, przed wireless-tools (jeśli obydwa są zainstalowane, w pierwszej kolejności używany jest wireless-tools).

Listing 2: Konfiguracja /etc/conf.d/net dla wpa_supplicant

```
# wpa_supplicant będzie użyty przed wireless-tools
modules=( "wpa_supplicant" )

# Bardzo istotne jest, aby wskazać wpa_supplicant który sterownik
# powinien zostać użyty, gdyż na obecnym etapie rozwoju nie jest jeszcze
# najlepszy w samodzielnym zgadywaniu
wpa_supplicant_eth0="-Dmadwifi"
```

Uwaga: Jeżeli używany jest sterownik host-ap, należy ustawić tryb zarządzania w karcie, zanim zacznie poprawnie współpracować z wpa_supplicant. Aby to osiągnąć, można użyć iwconfig_eth0="mode managed" w /etc/conf.d/net.

Nie było to trudne, prawda? Nadal jednak trzeba skonfigurować wpa_supplicant samo w sobie, co jest dość trudne w zależności od tego jak bezpieczne są access pointy, z którymi następuje połaczenie. Poniższy przykład jest uproszczoną wersją z pliku /etc/wpa_supplicant.conf.example, który pochodzi z wpa_supplicant.

Listing 3: Przykładowy /etc/wpa_supplicant.conf

```
# Zmiana poniższej linijki może spowodować, że wpa_supplicant nie będzie działać
ctrl_interface=/var/run/wpa_supplicant

# Należy być pewnym, że tylko root ma dostęp do konfiguracji WPA
ctrl_interface_group=0

# Niech wpa_supplicant zajmie się wyszukiwaniem i ustawianiem AP
ap_scan=1
```

```

# Prosty przykład: WPA-PSK, PSK oraz hasło w ASCII umożliwiają poprawną autoryzację
network={
    ssid="proste"
    psk="bardzo tajne hasło"
    # Im wyższy priorytet, tym wcześniej zostanie dopasowane
    priority=5
}

# Podobne jak poprzednie, ale będzie dokonane skanowanie SSID-ów (dla
# AP, które nie wysyłają swojego SSID-a)
network={
    ssid="drugi ssid"
    scan_ssid=1
    psk="bardzo tajne hasło"
    priority=2
}

# Jedynie WPA-PSK jest używane. Dowolna kombinacja hasła jest akceptowana
network={
    ssid="przykład"
    proto=WPA
    key_mgmt=WPA-PSK
    pairwise=CCMP TKIP
    group=CCMP TKIP WEP104 WEP40
    psk=06b4be19da289f475aa46a33cb793029d4ab3db7a23ee92382eb0106c72ac7bb
    priority=2
}
# Połączenie bez szyfrowania (brak WPA, brak IEEE 802.1X)
network={
    ssid="test-bez-szyfrowania"
    key_mgmt=NONE
}
# Połączenie ze współdzielonym WEP (brak WPA, brak IEEE 802.1X)
network={
    ssid="test-statycznego-wep"
    key_mgmt=NONE
    wep_key0="abcde"
    wep_key1=0102030405
    wep_key2="1234567890123"
    wep_tx_keyidx=0
    priority=5
}
# Połączenie ze współdzielonym WEP z kluczem (brak WPA, brak IEEE
# 802.1X) używając autoryzację ze współdzielonym kluczem IEEE 802.11
network={
    ssid="test2-statycznego-wep"
    key_mgmt=NONE
    wep_key0="abcde"
    wep_key1=0102030405
    wep_key2="1234567890123"
    wep_tx_keyidx=0
    priority=5
    auth_alg=SHARED
}
# Sieć IBSS/ad-hoc z WPA-None/TKIP
network={
    ssid="test adhoc"
    mode=1
    proto=WPA
    key_mgmt=WPA-NONE
    pairwise=None
    group=TKIP
    psk="tajne hasło"
}

```

4.c. Narzędzia do sieci bezprzewodowych

Wstępna konfiguracja i tryb zarządzany

[Wireless Tools](#) posiadają podstawowe metody na konfigurację podstawowych interfejsów sieci bezprzewodowych aż do ustawień poziomu zabezpieczeń WEP. Mimo, że WEP to dość słaba metoda zabezpieczeń, jest najczęściej stosowana.

Konfiguracje Wireless Tools są kontrolowane przy pomocy kilku głównych zmiennych. Przykładowa konfiguracja poniżej powinna opisać wszystko co jest potrzebne. Jedyne o czym należy pamiętać, to fakt, że dana konfiguracja nie oznacza "połącz się z najlepszym nieszyfrowanym access pointem" - zawsze nastąpi próba połączenia z czymkolwiek.

Listing 4: Instalacja wireless-tools

```
# emerge net-wireless/wireless-tools
```

Uwaga: Mimo, że ustawienia sieci bezprzewodowych można trzymać w /etc/conf.d/wireless, my radzimy przetrzymać je w /etc/conf.d/net.

Ważne: Koniecznie należy zatrzymać konfigurację opisującą nazwy zmiennych.

Listing 5: Przykładowe ustawienia iwconfig w /etc/conf.d/net

```
# W pierwszej kolejności zostanie użyte iwconfig przed wpa_supplicant
modules=( "iwconfig" )

# Konfiguracja dla Access Pointów nazwanych ESSID1 oraz ESSID2
# Można skonfigurować do 4 kluczy WEP, ale tylko jeden może być aktywny
# w danym momencie, tak aby domyślny indeks [1] był ustawiony na klucz [1] i z
# powrotem aby ustawić klucz na [1]. Dokonuje się tego w przypadku gdy
# ESSID został skonfigurowany dla kluczy WEP innych niż 1.
# Poprzedzając klucz przy pomocy s: oznacza że jest to klucz w ASCII, w innym
# przypadku jest to klucz w HEX
# enc open oznacza otwarte bezpieczeństwo (najbezpieczniejsze)
# enc restricted oznacza zastrzeżone bezpieczeństwo (mniej bezpieczne)
key_ESSID1="[1] s:twojklucz key [1] enc open"
key_ESSID2="[1] aaaa-bbbb-cccc-dd key [1] enc restricted"

# Poniższe zadziała tylko gdy będą poszukiwane dostępne Access Pointy

# Niekiedy widocznych jest więcej Access Pointów, więc należy
# zdefiniować który jest preferowany i w jakiej kolejności należy się łączyć
preferred_aps=( "ESSID1" "ESSID2" )
```

Konfiguracja wyboru punktów dostępu

Można dodać kilka opcji, aby lepiej skonfigurować wybór Access Pointów, jednakże na ogół nie są one wymagane.

Można zdecydować czy łączymy się tylko z preferowanymi Access Pointami czy nie. Domyślnie, jeśli żadne z ustawień nie zadziałają i można połączyć się do nieszyfrowanego Access Pointa, to nastąpi połączenie. Można to kontrolować przy pomocy zmiennej associate_order. Poniżej znajduje się tabela z wartościami oraz jak wpływają na kontrolę.

Wartość	Opis
any	Domyślne zachowanie
preferredonly	Będzie możliwa połączenie się jedynie z AP z listy preferowanych
forcepreferred	Będą dokonywane połączenia z AP w preferowanej kolejności jeśli nie zostały one odnalezione w skanowaniu
forcepreferredonly	Nie będzie skanowania w poszukiwaniu AP - w zamian nastąpi próba połączenia się z każdym w kolejności
forceany	Podobnie jak forcepreferred + połącz się z dowolnym dostępnym AP

Ostatecznie, istnieje możliwość wyboru blacklist_aps oraz unique_ap. blacklist_aps działa podobnie jak preferred_aps. unique_ap to wartość tak lub nie która wskazuje czy drugi interfejs bezprzewodowy może połączyć się do tego samego Access Pointa co pierwszy interfejs.

Listing 6: przykład z blacklist_aps oraz unique_ap

```
# Niekiedy nie powinno nastąpić połączenie do poszczególnych access pointów
blacklist_aps=( "ESSID3" "ESSID4" )

# Jeżeli jest więcej niż jedna karta bezprzewodowa, można określić czy
# powinno nastąpić połączenie każdej karty z tym samym punktem dostępu czy nie
# Wartości to "yes" (tak) lub "no" (nie)
# Domyślnie jest "yes"
unique_ap="yes"
```

Tryb Ad-Hoc oraz Zarządzany

Jeżeli zachodzi potrzeba ustawienia własnego węzła Ad-Hoc w przypadku braku możliwości połączenia się z jakimkolwiek Access Pointem, to również jest to możliwe.

Listing 7: Przenieś się na tryb ad-hoc

```
adhoc_essid_eth0="Ten Węzeł Adhoc"
```

A co z połączaniami do sieci Ad-Hoc lub działaniem w trybie zarządcy, aby stać się Access Pointem? Poniżej znajduje się konfiguracja do tego! Może zajść potrzeba, aby ustawić klucze WEP pokazane powyżej.

Listing 8: Przykładowa konfiguracja ad-hoc/master

```
# Ustawienie trybu - może być zarządzany (domyślnie), ad-hoc lub
# zarządcy. Nie wszystkie sterowniki umożliwiają skorzystanie z tych trybów
mode_eth0="ad-hoc"

# Ustawienie ESSID interfejsu
# W trybie zarządzanym, zmusza to interfejs do połączenia się z wyznaczonym
# ESSIESSID-em niczym innym
essid_eth0="Ten węzeł Adhoc"

# Jeśli nie zostanie wyznaczony żaden inny, zostanie użyty 3.
channel_eth0="9"
```

Ważne: Poniżej znajduje się dosłowny wycinek z BSD wavelan documentation, który znajduje się w zasobach dokumentacji NetBSD. Jest 14 kanałów do wyboru. Kanały od 1 do 11 są legalne w Północnej Ameryce, kanały od 1 do 13 w większości Europy, kanały od 10 do 13 we Francji, a kanał 14 jest przeznaczony dla Japonii. W przypadku wątpliwości, należy odnieść się do dokumentacji znajdującej się przy karcie bezprzewodowej lub access pointie. Należy upewnić się, że wybrany kanał jest ten sam który jest obsługiwany przez access point (lub inna karta znajdująca się w sieci typu ad-hoc). Domyślnym kanałem dla kart w Północnej Ameryce oraz Europie jest kanał 3; domyślny dla kart francuskich jest 11, zaś dla tych sprzedawanych w Japonii to 14.

Rozwiązywanie problemów z Wireless Tools

Jest kilka zmiennych, których można użyć do ustawienia i uruchomienia sieci bezprzewodowej, pomimo napotkanych problemów. Poniżej znajduje się tabela, która przedstawia zmienne do wyprobowania.

Zmienna

Domyślna Opis

	wartość	
<code>iwconfig_ether0</code>		Szczegóły dotyczące iwconfig znajdują się w manualu iwconfig
<code>iwpriv_ether0</code>		Szczegóły dotyczące iwpriv znajdują się w manualu iwpriv
<code>sleep_scan_ether0</code>	0	Liczba sekund uśpienia przed rozpoczęciem skanowania. Jest to potrzebne dla tych sterowników, które potrzebują więcej czasu na uruchomienie zanim mogą zostać użyte.
<code>sleep_associate_ether0</code>	5	Liczba sekund oczekiwania na połączenie się interfejsu z Access Pointem przed przeniesieniem się na kolejny
		Niektóre sterowniki nie zerują adresów MAC przypisanych gdy zgubią adres lub następuje próba połączenia.
<code>associate_test_ether0</code>	MAC	Niektóre sterowniki nie zerują poziomu jakości gdy zgubią adres lub następuje próba połączenia. Prawidłowe wartości to MAC, quality lub all.
<code>scan_mode_ether0</code>		Niektóre sterowniki muszą dokonać skanowania w trybie ad-hoc, więc jeśli skanowanie nie działa, należy ustawić tutaj ad-hoc
<code>iwpriv_scan_pre_ether0</code>		Wysyła polecenia iwpriv do interfejsu przed skanowaniem. Więcej informacji znajduje się w man iwpriv.
<code>iwpriv_scan_post_ether0</code>		Wysyła polecenia iwpriv do interfejsu po skanowaniu Więcej informacji znajduje się w man iwpriv.

4.d. Definiowanie konfiguracji sieci w zależności od ESSID

Niekiedy po połączeniu do ESSID1, zachodzi potrzeba otrzymania statycznego adresu IP, zaś w przypadku ESSID2 - dynamicznego przez DHCP. Większość zmiennych może być ustawianych w zależności od ESSIDA. Poniżej jest opisane jak tego dokonać.

Uwaga: Poniższe działają jeśli używa się WPA Suplicant lub Wireless Tools.

Ważne: Koniecznie należy zatrzymać dokumentacji opisującej nazwy zmiennych.

Listing 9: nadpisywanie ustawień sieciowych w zależności od ESSIDA

```
config_ESSID1=( "192.168.0.3/24 brd 192.168.0.255" )
routes_ESSID1=( "default via 192.168.0.1" )

config_ESSID2=( "dhcp" )
fallback_ESSID2=( "192.168.3.4/24" )
fallback_route_ESSID2=( "default via 192.168.3.1" )

# Można skonfigurować serwery nazw i inne rzeczy
# NOTATKA: DHCP to przepisze, chyba że zostanie ustawione inaczej
dns_servers_ESSID1=( "192.168.0.1" "192.168.0.2" )
dns_domain_ESSID1="some.domain"
dns_search_domains_ESSID1="szukaj.tej.domeny szukaj.tamtej.domeny"

# Nadpisuje się adres MAC adresem Access Pointa. Jest to przydatne gdy
# zachodzi potrzeba przemieszczania się między różnymi lokalizacjami, które
# posiadają ten sam ESSID
config_001122334455=( "dhcp" )
dhpcd_001122334455="-t 10"
dns_servers_001122334455=( "192.168.0.1" "192.168.0.2" )
```

5. Dodawanie możliwości

5.a. Standardowe zaczepy funkcji

Można zdefiniować cztery funkcje, które będą uruchamiane przy okazji operacji start/stop. Początkowo funkcje są uruchamiane razem z nazwą interfejsu tak, aby jedna funkcja mogła kontrolować wiele urządzeń.

Wartości jakie są zwracane po wykonaniu funkcji preup oraz predown powinny być równe 0 (sukces), aby powiadomić, że konfiguracja urządzenia mogła być kontynuowana. Jeżeli funkcja preup zwróci wartość różną od 0, konfiguracja zostanie przerwana. Jeżeli funkcja predown zwróci wartość różną od zera, to niemożliwa będzie dalsza dekonfiguracja urządzenia.

Wartości jakie są zwracane po wykonaniu funkcji postup oraz postdown są ignorowane, gdyż nic nie jest wykonywane w przypadku niepowodzenia.

`${IFACE}` jest ustawione jako nazwa urządzenia, które jest włączane/wyłączane. `${IFVAR}` to `${IFACE}` przekonwertowane do zmiennej akceptowanej przez bash.

Listing 1: Przykłady funkcji pre/post up/down

```

preup() {
    # Sprawdza czy istnieje link na interfejsie, który jest
    # uruchamiany. Ta opcja działa jedynie na kilku urządzeniach
    # sieciowych i wymaga, aby pakiet mii-diag był zainstalowany
    if mii-tool ${IFACE} 2> /dev/null | grep -q 'no link'; then
        ewarn "Nie ma linku na ${IFACE}, przerywam konfigurację"
        return 1
    fi

    # Sprawdza czy istnieje link na interfejsie przed jego
    # uruchomieniem. Ta opcja działa tylko na niektórych urządzeniach
    # sieciowych i wymaga, aby pakiet ethtool był zainstalowany
    if ethtool ${IFACE} | grep -q 'Link detected: no'; then
        ewarn "Brakuje linku na ${IFACE}, przerywam konfigurację"
        return 1
    fi

    # Należy pamiętać o zwróceniu wartości 0 w przypadku powodzenia
    return 0
}

predown() {
    # Domyslny skrypt sprawdza czy istnieje katalog główny NFS i
    # nie zezwala na wyłączenie interfejsów w takim przypadku. Należy
    # zauważyć, że jeśli określi się własną funkcję predown(), to właściwość
    # ta zostanie usunięta. Poniżej znajduje się ten skrypt, jeśli okazałoby
    # się, że jest nadal potrzebny...
    if is_net_fs /; then
        eerror "katalog główny jest zamontowany przez sieć -- nie można zatrzymać ${IFACE}"
        return 1
    fi

    # Należy pamiętać o zwróceniu wartości 0 w przypadku powodzenia
    return 0
}

postup() {
    # Ta funkcja zostanie użyta dla przykładu, aby zarejestrować
    # interfejs z usługą DNS. Inną możliwością jest np. wysłanie wiadomości
    # mailowej z informacją o uruchomieniu interfejsu
    return 0
}

postdown() {
    # Ta funkcja jest tu tylko dla zasady... Jeszcze nie
    # wiadomo co fajnego można z nią zrobić ;-)
    return 0
}

```

5.b. Funkcje dla sieci bezprzewodowych

Uwaga: Poniższe funkcje nie zadziałają z WPA Suplicant, ale zmienne \${ESSID} oraz \${ESSIDVAR} są dostępne w funkcji postup().

Można zdefiniować dwie funkcje, które zostaną uruchomione razem z powiązaną funkcją. Funkcje te są uruchamiane najpierw z nazwą interfejsu, aby jedna z nich mogła obsługiwać wiele urządzeń.

Zwracane wartości dla funkcji uruchamianych przed uruchomieniem urządzenia powinny być równe 0 (sukces), aby wskazać, że konfiguracja może być kontynuowana. Jeżeli zostanie zwrócona wartość różna od zera, konfiguracja zostanie przerwana.

Zwracane wartości dla funkcji preassociate są ignorowane gdyż i tak nie mają wpływu na dalszą konfigurację.

`\${ESSID}` jest równe wartości ESSID punktu dostępu z którym dokonywane jest połaczenie. `\${ESSIDVAR}` jest równe `\${ESSID}` przekonwertowane do zmiennej akceptowanej przez powłokę bash

Listing 2: Funkcje pre/post association

```
preassociate() {
    # Poniższe linijki dodają dwie zmienne leap_user_ESSID oraz
    # leap_pass_ESSID. Jeżeli obie są skonfigurowane dla ESSID-a z którym są
    # połączone, uruchamiany jest skrypt CISCO LEAP

    local user pass
    eval user=\"$${leap_user_${ESSIDVAR}}\`"
    eval pass=\"$${leap_pass_${ESSIDVAR}}\`"

    if [[ -n ${user} && -n ${pass} ]]; then
        if [[ ! -x /opt/cisco/bin/leapscript ]]; then
            eend "For LEAP support, please emerge net-misc/cisco-aironet-client-utils"
            return 1
        fi
        einfo "Waiting for LEAP Authentication on \"${ESSID//\\\\\\\\}\\\""
        if /opt/cisco/bin/leapscript ${user} ${pass} | grep -q 'Login incorrect'; then
            ewarn "Login Failed for ${user}"
            return 1
        fi
    fi
    return 0
}

postassociate() {
    # Ta funkcja jest tu tylko dla zasady... Jeszcze nie
    # wiadomo co fajnego można z nią zrobić ;-)

    return 0
}
```

Uwaga: \${ESSID} oraz \${ESSIDVAR} są niedostępne dla funkcji predown() oraz postdown().

6. Zarządzanie siecią

6.a. Zarządzanie siecią

Jeżeli komputer jest ciągle "w drodze", nie zawsze może być podłączony do sieci, czy też posiadać dostępu do access pointa. W takich przypadkach może okazać się, że pożądane by było, aby sieć włączała się automatycznie w momencie gdy możliwy jest do niej dostęp.

Poniżej można znaleźć opis kilku narzędzi, które mogą być w tym pomocne.

Uwaga: Ten dokument opisuje jedynie ifplugd, jednakże istnieją alternatywy, jak na przykład quickswitch.

6.b. ifplugd

ifplugd to demon, który uruchamia i zatrzymuje urządzenia sieciowe gdy kabel sieciowy jest wkładany lub wyjmowany z gniazda. Może również zarządzać przypisami do access pointów, gdy jakiś nowy pojawi się w zasięgu.

Listing 1: Instalacja ifplugd

```
# emerge sys-apps/ifplugd
```

Konfiguracja ifplugd jest bardzo prosta. Plik konfiguracyjny znajduje się w /etc/conf.d/ifplugd. man ifplugd pokaże co poszczególne opcje oznaczają.

Listing 2: Przykładowa konfiguracja ifplugd

```
# Definicja urządzeń do monitorowania
INTERFACES="eth0"

AUTO="no"
BEEP="yes"
IGNORE_FAIL="yes"
IGNORE_FAIL_POSITIVE="no"
IGNORE_RETVAL="yes"
POLL_TIME="1"
DELAY_UP="0"
DELAY_DOWN="0"
API_MODE="auto"
SHUTDOWN="no"
WAIT_ON_FORK="no"
MONITOR="no"
ARGS=""

# Dodatkowe parametry dla ifplugs dla wskazanego urządzenia. Należy
# zwrócić uwagę, że globalna zmienna jest ignorowana, jeśli zmienna szczegółowa
# jest ustawiona dla danego urządzenia.
MONITOR_wlan0="yes"
DELAY_UP_wlan0="5"
DELAY_DOWN_wlan0="5"
```

E. Dokumentacja dla Gentoo na stacjach roboczych

1. Instalacja i konfiguracja serwera X

1. Czym jest serwer X?

Środowisko graficzne kontra konsola

Przeciętny użytkownik boi się wpisywania poleceń. Znacznie prostsze wydaje mu się wskazanie celu i kliknięcie na nim odpowiednim przyciskiem. Czemu zatem nie pozwolić im w Gentoo pracować właśnie w ten sposób? Cóż, pozwalamy im na to :) Linux posiada wiele świetnie wyglądających interfejsów i środowisk, które każdy z użytkowników może zainstalować w celu usprawnienia pracy.

Jest to z pewnością jedna z największych niespodzianek dla nowych użytkowników, kiedy okazuje się, że graficzny interfejs, którym się posługują jest jedynie kolejną aplikacją działającą w systemie. Nie jest częścią jadra, ani żadnych innych ważnych wewnętrznych części systemu. Jest za to na pewno potężnym narzędziem znacznie poszerzającym graficzne możliwości każdego komputera.

W związku z tym, że standardy są bardzo ważną rzeczą, utworzono także standard rysowania i przemieszczania okien na ekranie, współpracy z użytkownikiem poprzez myszkę i klawiaturę oraz innych podstawowych aspektów działania takiego środowiska. Nazwano go systemem X Window, którego nazwę często skraca się do zwrotu X11 lub po prostu X. System ten jest wykorzystywany na platformach UNIX, Linux oraz na wielu innych do nich podobnych.

Opartą na standardzie X11 aplikacją, która pozwala użytkownikom Linuksa na korzystanie ze środowiska graficznego jest Xorg-X11, który został jakiś czas temu odłączony od projektu XFree86. Osoby rozwijające XFree86 zdecydowały się na zmianę licencji na niekompatybilną z GPL, dlatego zalecamy korzystanie z Xorg-X11. W oficjalnym drzewie Portage nie ma już ebuildów XFree86.

Projekt X.org

Projekt [X.org](#) to grupa osób, które stworzyły i zarządzają ogólnodostępną i posiadającą otwarte źródła implementacją systemu X11.

Xorg to interfejs pomiędzy sprzętem i programami graficznymi, które użytkownik zamierza uruchomić. Jest to ponadto aplikacja mogąca działać w sieci i umożliwiająca zdalne uruchamianie aplikacji graficznych z innych komputerów.

2. Instalowanie Xorg

Użycie emerge

Koniec wstępu, pora zajść się czymś poważnym. Aby zainstalować Xorg, wystarczy wpisać polecenie emerge xorg-x11. Instalowanie zajmuje trochę czasu, zalecamy cierpliwość i może zjedzenie jakiegoś ciepłego posiłku.

Przed instalacją Xorga należy skonfigurować dwie ważne zmienne w pliku /etc/make.conf.

Pierwsza zmienna nazywa się VIDEO_CARDS i zawiera nazwy sterowników do kart graficznych jakich zamierzamy używać. Najczęściej pojawiają się tu wpisy nvidia dla kart Nvidia i fglrx dla kart ATI. Są to firmowe sterowniki dostarczane przez producentów sprzętu. Jeśli zamiast tego zamierza używać się sterowników otwartych, należy wpisać do zmiennej nv zamiast nvidia i radeon zamiast fglrx. Użytkownicy sterownika nv muszą jednak pamiętać, że jego użycie oznacza brak jakiegokolwiek przyspieszenia sprzętowego. Zmienna VIDEO_CARDS może zawierać więcej nazw sterowników, ich lista powinna być rozdzielona spacjami.

Druga zmienna ma nazwę INPUT_DEVICES i jest wykorzystywana do zdefiniowania sterowników urządzeń wejścia jakich zamierza się używać. Zwykle wystarcza wpis o treści keyboard mouse.

Teraz należy zdecydować jakich sterowników zamierza się używać oraz dopisać je do odpowiednich zmiennych w pliku /etc/make.conf:

Listing 2.1: Przykładowe wpisy w pliku make.conf

```
(Obsługa klawiatury i myszki)
INPUT_DEVICES="keyboard mouse"
(Dla kart Nvidia)
VIDEO_CARDS="nvidia"
(LUB, dla kart ATI Radeon)
VIDEO_CARDS="fglrx"
```

Więcej informacji na temat konfiguracji kart Nvidia i ATI znajduje się w teksth Konfiguracja kart nVidia w Gentoo i Praca z kartami ATI w Gentoo. Można posłużyć się tymi tekstami w celu wybrania odpowiednich sterowników do posiadanej karty.

Uwaga: Sugerowane ustawienia dotyczą większości użytkowników, ale na pewno nie zadziałają wszędzie. W takim przypadku należy wpisać polecenie emerge -pv xorg-x11, sprawdzić wszystkie możliwe ustawienia zmiennych i wybrać odpowiednie dla posiadanego sprzętu sterowniki. W różnych wersjach Xorga na różnych architekturach flagi te są ustawiane dla emerge -pv xorg-server. Przykład dotyczy

architektury x86 i xorg-x11-7.0.

Listing 2.2: Wyświetlanie wszystkich dostępnych opcji

```
# emerge -pv xorg-x11
```

These are the packages that would be merged, in order:

Calculating dependencies... done!

```
[ebuild R ] x11-base/xorg-x11-7.0-r1 USE="-3dfx" INPUT_DEVICES="keyboard
mouse -acecad -aiptek -calcomp -citron -digitaledge -dmc -dynapro -elo2300
-elographics -evdev -fpit -hyperpen -jamstudio -joystick -magellan -magictouch
-microtouch -mutouch -palmax -penmount -spaceorb -summa -synaptics -tek4957
-ur98 -vmmouse -void -wacom" VIDEO_CARDS="nvidia -apm -ark -chips -cirrus
-cyrix -dummy -fbdev -fglrx -glint -i128 -i740 -i810 -imstt -mach64 -mga
-neomagic -nsc -nv -r128 -radeon -rendition -s3 -s3virge -savage -siliconmotion
-sis -sisusb -tdfx -tga -trident -tseng -v4l -vesa -via -vmware -voodoo" 0kB
```

Po ustawieniu zmiennych można przystąpić do instalacji pakietu Xorg.

Listing 2.3: Instalowanie Xorg

```
# emerge xorg-x11
```

Po zakończeniu procesu instalacji należy odświeżyć kilka zmiennych środowiskowych przed przejściem do konfiguracji X. Służą do tego polecenia env-update i source /etc/profile. W żaden sposób nie zakłócają one normalnej pracy systemu.

Listing 2.4: Odświeżanie zmiennych środowiskowych

```
# env-update
# source /etc/profile
```

3. Konfigurowanie Xorg

Plik xorg.conf

Konfiguracja Xorg znajduje się w pliku o nazwie xorg.conf w katalogu /etc/X11. Pakiet Xorg-X11 zawiera przykładowy plik konfiguracyjny, /etc/X11/xorg.conf.example, który można wykorzystać jako podstawę do tworzenia własnej konfiguracji. Znajduje się w nim mnóstwo komentarzy, a jeszcze więcej informacji o jego składni można znaleźć na jego stronie man.

Listing 3.1: Czytanie strony man xorg.conf

```
# man 5 xorg.conf
```

Życzymy miłej lektury wszystkim, którzy przystąpili do czytania tego manuala, a z resztą grupy przechodzimy do metod automatycznego generowania tego pliku.

Domyślnie: Automatyczna konfiguracja xorg.conf

Xorg jest w stanie samodzielnie wykryć parametry pracy. W większości przypadków wystarcza później zmiana kilku linii dotyczących rozdzielcości, aby poprawnie uruchomić środowisko. Więcej informacji o lepszym dostosowywaniu pliku konfiguracyjnego do swoich potrzeb można znaleźć w zasobach na końcu tego tekstu. Po pierwsze utworzymy działający plik konfiguracyjny Xorg.

Listing 3.2: Generowanie pliku xorg.conf

```
# Xorg -configure
```

Naprawdę warto przeczytać ostatnie linie wyświetcone na ekranie po tym jak Xorg zakończy sprawdzanie sprzętu. Jeśli nie uda się to w którymś momencie, konieczne będzie ręczne napisanie pliku xorg.conf. Jeśli jednak automatyczna konfiguracja nie zawiedzie, zostanie utworzony plik /root/xorg.conf.new gotów do testowania. I właśnie teraz zajmiemy się jego testowaniem.

Listing 3.3: Testowanie pliku xorg.conf

```
# X -config /root/xorg.conf.new
```

Jeśli wszystko pójdzie dobrze, pojawi się prosty czarno-biały obraz. Można sprawdzić czy myszka działa poprawnie oraz czy rozdzielcość jest odpowiednia. Nie łatwo wprawdzie zgadnąć jaka dokładnie została ustawiona, ale na pewno zauważa się kiedy jest ona za niska. Środowisko można zamknąć w dowolnym momencie za pomocą kombinacji klawiszy Ctrl-Alt-Backspace.

Alternatywnie: Półautomatyczne generowanie xorg.conf

Xorg zawiera narzędzie o nazwie xorgconfig, które może przeprowadzić użytkownika przez proces konfiguracji zadając mu serię pytań dotyczących komputera (model karty graficznej, układ klawiszy klawiatury...). Program opierając się na podanych odpowiedziach utworzy odpowiedni plik xorg.conf.

Listing 3.4: Półautomatyczne generowanie pliku xorg.conf

```
# xorgconfig
```

Inne narzędzie, również wchodzące w skład Xorg, xorgcfg, najpierw uruchamia skrypt Xorg -configure, a następnie uruchamia serwer X w celu przeprowadzenia dodatkowej konfiguracji.

Listing 3.5: Użycie xorgcfg

```
# xorgcfg
(Jeśli X nie uruchamia się prawidłowo lub konfiguracja się nie udaje
należy użyć poniższego polecenia)
# xorgcfg -textmode
```

Kopiowanie xorg.conf

Jeśli nie chcemy przy każdym uruchomieniu serwera X wpisywać polecenia X -config, musimy przekopiować plik xorg.conf.new w miejsce /etc/X11/xorg.conf. Dzięki temu znacznie uprościmy uruchamianie serwera, wystarczające będzie wpisanie polecenia X lub startx.

Listing 3.6: Kopiowanie xorg.conf

```
# cp /root/xorg.conf.new /etc/X11/xorg.conf
```

Używanie startx

Polecenie startx służy do uruchamiania serwera X. Jest to skrypt uruchamiający sesję X, która zajmuje się uruchamianiem serwerów X, a na nich innych aplikacji graficznych. Decyzja na temat tego jaka aplikacja graficzna ma zostać uruchomiona jest podejmowana na zasadach następującego algorytmu:

- Jeśli w katalogu domowym użytkownika znajduje się plik o nazwie .xinitrc to zostaną uruchomione polecenia znajdujące się w nim
- Jeśli nie istnieje to zostanie odczytana zawartość zmiennej XSESSION i zostaną uruchomione sesje z katalogu /etc/X11/Sessions/. Zmienną XSESSION można skonfigurować w pliku /etc/rc.conf, dzięki czemu stanie się ona domyślną dla wszystkich użytkowników danego systemu.
- Jeśli wszystko powyższe zawiedzie, zostanie uruchomiony domyślny prosty menedżer okien o nazwie twm

Listing 3.7: Uruchamianie X

```
# startx
```

Jeśli coś co pojawia się przed oczyma użytkownika po wydaniu tego polecenia jest bardzo brzydkie to najprawdopodobniej jest to właśnie twm. W celu zakończenia sesji twm należy wpisać polecenie exit lub wcisnąć Ctrl-D w jednym z terminali, które pojawią się po uruchomieniu serwera. Można również bezpośrednio wyłączyć X session kombinacją klawiszy Ctrl-Alt-Backspace. Jest to nie do końca czysta metoda, ale na pewno też nic nie zepsuje.

4. Dodatkowa konfiguracja xorg.conf

Ustawianie rozdzielczości

Jeśli pojawia się nieodparte wrażenie, że rozdzielcość ekranu jest niewłaściwa, trzeba sprawdzić dwie linie w pliku konfiguracyjnym. Przede wszystkim ważna jest sekcja Screen zawierająca spis wszystkich dostępnych rozdzielcości, o ile jakieś są dostępne, na których może działać serwer X. Domyślnie nie ma w niej żadnego wpisu, a Xorg wylicza rozdzielcość na podstawie innej sekcji, zatytułowanej Monitor.

Xorg sprawdza wartości HorizSync i VertRefresh w sekcji Monitor i na ich podstawie wylicza właściwą rozdzielcość. Na razie pozostawimy te ustawienia tak jak są. Jeśli znajdujące się nieco niżej proponowane zmiany w sekcji Screen nie dadzą oczekiwanych rezultatów, trzeba będzie sprawdzić w instrukcji monitora jego dokładne dane techniczne i wpisać do pliku konfiguracyjnego ich poprawne wartości. Jest dostępne narzędzie, które zostało napisane dla wyszukiwania wartości HorizSync i VertRefresh, nazywa się sys-apps/ddcxinfo-knoppix.

Ostrzeżenie: Nie należy wpisywać obu wartości dotyczących parametrów technicznych na wyczucie, bez skonsultowania się z odpowiednią specyfikacją techniczną monitora. Ich złe ustawienie prowadzi do błędów związanych ze zlym zakresem synchronizacji i może prowadzić nawet do spalenia się monitora.

Kolejny krok to zmiana rozdzielcości. W poniższym przykładzie wyciętym z pliku /etc/X11/xorg.conf dodaliśmy wpisy Modes i DefaultDepth, dzięki czemu serwer X będzie się uruchamiał z 24-bitową głębią kolorów w rozdzielcości 1024x768. Co do poniższego wycinka nie należy go przepisywać dosłownie, wartości niemal na pewno będą się różniły od tych pożądanych w konkretnym przypadku.

Listing 4.1: Zmiany w sekcji Screen pliku /etc/X11/xorg.conf

```
Section "Screen"
    Identifier "Default Screen"
    Device    "S3 Inc. ProSavage KN133 [Twister K]"
    Monitor   "Generic Monitor"
    DefaultDepth 24
    # Wycinamy kawałek tekstu w celu zwiększenia czytelności
    SubSection "Display"
        Depth    24
        Modes   "1024x768"
    EndSubSection
EndSection
```

Na koniec testujemy, czy zmiany dokonane w ustawieniach rozdzielcości były prawidłowe, poprzez uruchomienie serwera X (polecienniem startx).

Konfigurowanie klawiatury

Aby skonfigurować w X ustawienie dotyczące międzynarodowej klawiatury należy udać się do sekcji InputDevice pliku konfiguracyjnego i dodać w niej opcję XkbLayout z kodem pożądanego układu klawiatury. W przykładzie pokażemy jak skonfigurować polski układ klawiszy. Wybranie innego jest prostą sprawą, wystarczy wpisać w odpowiednim miejscu inny kod państwa.

Listing 4.2: Zmiana konfiguracji układu klawiszy

```
Section "InputDevice"
    Identifier "Generic Keyboard"
    Driver     "keyboard"
    Option    "CoreKeyboard"
    Option    "XkbRules" "xorg"
    Option    "XkbModel" "pc105"
    Option    "XkbLayout" "pl"
EndSection
```

Konfigurowanie myszki

Jeśli myszka nie działa to po pierwsze trzeba sprawdzić czy została wykryta przez jądro. Pliki urządzeń myszek to /dev/input/mouse0 (lub /dev/input/mice - jeśli posiada się ich kilka). W niektórych przypadkach korzysta się także z /dev/psaux. W każdym z przypadków zmieniająca się w trakcie poruszania myszką zawartość pliku będzie świadczyła o tym, że urządzenie działa poprawnie. Sama treść nie ma znaczenia, ważne, że coś się zmienia przy poruszaniu myszką. Aby zakończyć test wystarczy wcisnąć Ctrl-C.

Listing 4.3: Sprawdzanie plików urządzenia

```
# cat /dev/input/mouse0
(Na koniec należy wcisnąć CTRL-C)
```

Jeśli myszka uparcie nie jest wykrywana, należy sprawdzić czy wszystkie niezbędne moduły są załadowane.

Kiedy myszka wreszcie zostanie załadowana należy wypełnić odpowiedni wpis dla nazwy urządzenia w sekcji InputDevice. W poniższym przykładzie skonfigurujemy dodatkowo jeszcze dwie opcje: Protocol (czyli nazwę protokołu jaka ma być używana - dla większości użytkowników poprawny jest wpis PS/2 lub IMPS/2) oraz ZAxisMapping (dającą możliwość korzystania z kółka myszki (jeśli takie się posiada)).

Listing 4.4: Konfigurowanie myszki w Xorg

```
Section "InputDevice"
    Identifier  "TouchPad Mouse"
    Driver      "mouse"
    Option      "CorePointer"
    Option      "Device"      "/dev/input/mouse0"
    Option      "Protocol"    "IMPS/2"
    Option      "ZAxisMapping" "4 5"
EndSection
```

Na koniec należy po prostu wpisać polecenie startx i cieszyć się jego rezultatami :). Gratulujemy, serwer X działa teraz prawidłowo. Kolejnym krokiem jest usunięcie tego brzydkiego domyślnego menedżera okien i zainstalowanie jakiegoś znacznie bardziej użytecznego programu tego typu (czasem nawet niemal kompletnego środowiska) takiego jak GNOME czy KDE. Nie jest to już jednak tematem tego tekstu.

2. Instalacja i konfiguracja kart nVidia

1. Wprowadzenie

nVidia dostarcza sterowników poprawiających wydajność i zapewniających pełne wsparcie dla akceleracji 3D. Sterowniki te w Gentoo zostały podzielone na dwa pakiety: nvidia-kernel i nvidia-glx.

nvidia-kernel to sterownik jądra, który zapewnia niskopoziomową komunikację ze sprzętem video. Instaluje moduł jądra o nazwie nvidia, który tą obsługą się zajmuje i musi być załadowany za każdym razem kiedy zechcemy skorzystać ze sterowników nvidia.

Razem ze sterownikiem jądra muszą zostać zainstalowane biblioteki zawarte w pakiecie nvidia-glx. Są one wykorzystywane przez serwer X do renderowania grafiki, podczas gdy oryginalny sterownik (nvidia-kernel) jest wykorzystywany do komunikacji ze sprzętem.

2. Konfiguracja karty

Konfiguracja jądra

Tak jak wspominaliśmy wcześniej sterowniki nVidii zawierają moduły jądra, które musimy włączyć do jądra. By tego dokonać jądro musi obsługiwać ładowanie modułów. Jeśli użyto genkernel, aby skonfigurować jądro wszystko powinno być odpowiednio ustawione. Jeśli nie, należy sprawdzić dwa razy konfigurację jądra i upewnić się, że mamy to włączone:

Listing 2.1: Włączanie obsługi ładowania modułów jądra

```
Loadable module support --->
 [*] Enable loadable module support
```

Musimy także włączyć obsługę Memory Type Range Register:

Listing 2.2: Włączanie MTRR

```
Processor and Features --->
 [*] MTRR (Memory Type Range Register) support
```

Potrzebne będzie także dodanie wsparcia dla agpgart przez wkompilowanie bezpośrednio w jądro, lub jako moduł:

Listing 2.3: Włączanie agpgart

```
Device Drivers --->
Character devices --->
<*> /dev/agpgart (AGP Support)
```

Uwagi do architektur

Ważne: Dla procesorów x86 oraz AMD64, sterownik dostarczany przez jądro konfliktuje z binarnym sterownikiem dostarczonym przez producenta. Podczas komplikacji dla tych architektur należy całkowicie usunąć wsparcie dla sterowników dostarczanych przez jądro, tak jak to pokazano poniżej:

Listing 2.4: Usunięcie sterownika z jądra

```
Device Drivers --->
Graphics Support --->
< > nVidia Framebuffer Support
< > nVidia Riva support
```

Dobrą alternatywą dla ramki bufora jest VESA:

Listing 2.5: Włączenie wsparcia dla VESA

```
Device Drivers --->
Graphics Support --->
<*>   VESA VGA graphics support
```

Następnie, w sekcji "VESA driver type" należy wybrać typ sterownika: vesafb lub vesafb-tng: (vesafb-tng nie jest dostępny dla posiadaczy procesorów AMD64, przyp. tłum.)

Listing 2.6: Wybór typu ramki bufora

```
( ) vesafb
(X) vesafb-tng
```

Więcej informacji można znaleźć w /usr/src/linux/Documentation/fb/vesafb.txt w przypadku korzystania z vesafb lub w dokumentacji dla konkretnej ramki bufora w katalogu /usr/src/linux/Documentation/fb/.

Kontynuowanie konfiguracji jądra

Moduły i biblioteki nVidii to dwa pakiety nvidia-glx i nvidia-kernel. Pierwszy to biblioteki GLX dla X11, a drugi to moduły jądra.

Ebuild nvidia-kernel automatycznie wykrywa wersję używanego jądra w oparciu o plik /usr/src/linux. W związku z tym należy się upewnić, że jest on prawidłowym dowiązaniem symbolicznym do źródeł używanego jądra.

Jeśli korzysta się z jądra gentoo-sources-2.6.11-r6, to katalog /usr/src powinien wyglądać tak:

Listing 2.7: Sprawdzanie dowiązania do źródeł jądra

```
# cd /usr/src
# ls -l
(Sprawdzanie czy dowiązanie o nazwie linux wskazuje właściwy katalog)
lrwxrwxrwx 1 root root 22 Apr 23 18:33 linux -> linux-2.6.11-gentoo-r6
drwxr-xr-x 4 root root 120 Apr  8 18:56 linux-2.4.26-gentoo-r4
drwxr-xr-x 18 root root 664 Dec 31 16:09 linux-2.6.10
drwxr-xr-x 18 root root 632 Mar  3 12:27 linux-2.6.11
drwxr-xr-x 19 root root 4096 Mar 16 22:00 linux-2.6.11-gentoo-r6
```

W powyższym przykładzie widać, że dowiązanie o nazwie linux wskazuje na jądro linux-2.6.11-gentoo-r6

Jeśli dowiązanie nie jest prawidłowe to należy je poprawić w następujący sposób:

Listing 2.8: Tworzenie lub uaktualnianie dowiązania symbolicznego /usr/src/linux

```
# cd /usr/src
# ln -snf linux-2.6.11-gentoo-r6 linux
```

Opcjonalnie: Sprawdzanie wsparcia dla karty

Uwaga: Niestety, niektóre stare karty graficzne nie są dłużej wspierane przez nowe wersje nvidia-glx i nvidia-kernel. nVidia dostarcza listę wpieranych kart. Proszę sprawdzić tę listę przed instalacją sterowników.

Poniżej znajduje się lista starych **niewspieranych** już kart graficznych:

Listing 2.9: Niewspierane dłużej karty

```
TNT2
TNT2 Pro
TNT2 Ultra
TNT2 Model 64 (M64)
TNT2 Model 64 (M64) Pro
Vanta
Vanta LT
GeForce 256
GeForce DDR
GeForce2 GTS
GeForce2 Pro
GeForce2 Ti
GeForce2 Ultra
GeForce2 MX Integrated graphics
Quadro
Quadro2 Pro
Quadro2 EX
```

Jeśli jest się posiadaczem jednej z powyższych kart, należy zamaskować nowszą wersję sterowników nVidia i zainstalować starszą.

Listing 2.10: Maskowanie nowszych sterowników

```
# echo ">media-video/nvidia-kernel-1.0.6629-r5" >> /etc/portage/package.mask
# echo ">media-video/nvidia-glx-1.0.6629-r7" >> /etc/portage/package.mask
```

Instalacja sterowników

Następnie pora zainstalować pakiety nvidia-kernel i nvidia-glx. W związku z tym, że nvidia-kernel jest zależnością nvidia-glx wystarczy zainstalować tylko ten drugi pakiet.

Listing 2.11: Instalacja modułów nVidii

```
# emerge nvidia-glx
```

Ważne: Za każdym razem gdy komplujemy jądro musimy ponownie uruchomić polecenie emerge nvidia-kernel w celu ponownego zainstalowania modułów. Zmiany jądra nie mają wpływu na pakiet nvidia-glx.

Gdy instalacja się zakończy wpisujemy modprobe nvidia, aby załadować moduł jądra do pamięci:

Listing 2.12: Ladowanie modułu jądra

```
# modprobe nvidia
```

Dobrym pomysłem i wygodnym rozwiązaniem jest, aby działało się to automatycznie przy każdym uruchomieniu systemu. Aby to osiągnąć wystarczy dodać linijkę nvidia do pliku /etc/modules.autoload.d/kernel-2.6 (lub kernel-2.4). Później należy wykonać polecenie modules-update.

Ważne: Jeśli agpgart zostało skompilowane jako moduł, będzie trzeba dodać je do /etc/modules.autoload.d/kernel-2.6 (lub do kernel-2.4, w zależności od wersji jądra).

Listing 2.13: Uruchamianie modules-update

```
# modules-update
Konfiguracja Serwera X
```

Po instalacji odpowiednich sterowników należy skonfigurować serwer X (XFree86 lub Xorg), aby używał sterownika nvidia zamiast domyślnego nv.

Otwieramy /etc/X11/xorg.conf ulubionym edytorem (np. nano lub vim) i przechodzimy do sekcji Device. Należy tam odpowiednio zmienić linię Driver:

Listing 2.14: Zamiana nv na nvidia w konfiguracji serwera X

```
Section "Device"
    Identifier "nVidia Inc. GeForce2"
    Driver      "nvidia"
    VideoRam   65536
EndSection
```

W sekcji Module należy upewnić się, że moduł glx będzie ładowany i że moduł dri nie będzie wczytywany:

Listing 2.15: Uaktualnianie sekcji Module

```
Section "Module"
    (...)

    # Load "dri"
    Load  "glx"
    (...)

EndSection
```

Następnie w sekcji Screen należy upewnić się, że dyrektywa DefaultDepth jest ustawiona na 16 lub 24 lub w podsekcjach Display mamy Depth ustawione tylko i wyłącznie na 16 lub 24. Bez tego rozszerzenia nvidia-glx nie zostaną uruchomione.

Listing 2.16: Uaktualnianie sekcji Screen

```
Section "Screen"
    (...)

    DefaultDepth 16
    Subsection "Display"
        (...)

EndSection
```

Na koniec uruchamiamy polecenie eselect, co sprawi, że serwer X będzie używał bibliotek GLX nVidii:

Listing 2.17: Uruchamianie opengl-update

```
# eselectopengl set nvidia
```

Sprawdzanie karty

Aby sprawdzić kartę nVidia, należy uruchomić serwer X i wpisać polecenie glxinfo | grep direct. Powinna pokazać się informacja, że bezpośrednie renderowanie (ang. direct rendering) jest włączone:

Listing 2.18: Sprawdzanie statusu bezpośredniego renderowania

```
$ glxinfo | grep direct
direct rendering: Yes
```

Do sprawdzania ilości FPS korzystamy z programu glxgears.

Sprawdzanie karty

W celu sprawdzenia czy sterowniki zostały zainstalowane prawidłowo, należy wpisać polecenie `glxinfo | grep direct`. Powinno ono poinformować o tym, że działa bezpośrednie generowanie grafiki:

Listing 2.19: Sprawdzanie karty

```
$ glxinfo | grep direct
direct rendering: Yes
```

W celu sprawdzenia ilości FPS należy uruchomić `glxgears`.

Włączanie obsługi nvidia

Niektóre programy, takie jak np. `mplayer` czy `xine-lib` wykorzystują flagę `USE` o nazwie "nvidia", która włącza `XvMCNVIDIA`, przydatną podczas oglądania filmów w bardzo dużej rozdzielcości. Aby ją aktywować należy dodać wpis "nvidia" do listy zmiennych w pliku `/etc/make.conf` lub dodać jako flagę wyłącznie dla `media-video/mplayer` lub `media-libs/xine-lib` w pliku `/etc/portage/package.use`.

Na koniec należy wpisać polecenie `emerge -uD --newuse world`, które przebuduje wszystkie programy, tak by mogły korzystać z nowych ustawień flag `USE`.

Korzystanie z narzędzi do konfiguracji NVidia

Od wydania przez NVidię wersji 1.0.6106, sterowniki dostarczane są z narzędziem konfiguracyjnym. Narzędzie to pozwala nam na zmianę ustawień graficznych bez potrzeby restartowania serwera X. Dostępne są one w Portage jako `media-video/nvidia-settings`.

3. Rozwiązywanie problemów

Co zrobić, aby działała grafika 2D na komputerach z 4GB (lub więcej) pamięci

Jeśli pojawią się problemy z przyspieszeniem grafiki 2D prawdopodobnie nie można zestawić zakresu kombinacji zapisu (ang. write-combining ranges) z MTRR. Aby się upewnić, należy sprawdzić zawartość `/proc/mtrr`:

Listing 3.1: Sprawdzanie, czy mamy włączone kombinacje zapisu

```
# cat /proc/mtrr
```

W liniach powinno być napisane "write-back" lub "write-combining". Jeśli tak nie jest to trzeba przestawić ustawienie w BIOS-ie aby to naprawić.

Uruchamiamy ponownie komputer i wchodzimy do BIOS-u, znajdujemy tam ustawienia MTRR (prawdopodobnie w "Ustawieniach CPU") i zmieniamy ustawienie z "ciągłe" (ang. continuous) na "dyskretne" (ang. discrete), a następnie zapisujemy i uruchamiamy Linuksa. Zniknie wpis "uncachable", a przyspieszenie 2D zacznie działać bez zarzutu.

Otrzymuję ostrzeżenia o nieobsługiwany rozmiarze stosu (4K)

Pakiety `nvidia-kernel` w wersji starszej od 1.0.6106 nie posiadają obsługi stosów mniejszych niż o rozmiarze 8K. Nowsze jądra (wersje 2.6.6 i nowsze) posiadają także obsługę 4K rozmiaru stosu. Nie należy zaznaczać 4K rozmiaru stosu w konfiguracji jądra jeśli używamy starszej wersji pakietu `nvidia-kernel`. Opcję tą można znaleźć w sekcji Kernel Hacking.

Kiedy ładuję moduł dostaję ostrzeżenie, że nie ma takiego urządzenia ("no such device").

Z reguły zdarza się to w sytuacji gdy nie mamy odpowiedniej karty graficznej. Należy się upewnić, że posiadamy kartę graficzną opartą o GPU NVidia (możemy tego dokonać przy użyciu polecenia `lspci`).

Jeśli jesteśmy pewni, że posiadamy kartę graficzną NVidia, należy sprawdzić ustawienia BIOS-u i upewnić się, że pozycja Assign IRQ to VGA jest poprawnie ustawiona.

Pojawia się błąd "no screens found", a w logach komunikat "Failed to initialize the NVIDIA kernel module!"

Zapewne brakuje plików urządzeń `/dev/nvidia*`. Tworzy się je za pomocą skryptu `NVmakedevices.sh`:

Listing 3.2: Tworzenie brakujących plików urządzeń

```
# /sbin/NVmakedevices.sh
```

Jeśli brak urządzenia `/dev/nvidia*` powtarza się po każdym ponownym uruchomieniu komputera, najprawdopodobniej system udev nie tworzy odpowiednich odpowiednich plików urządzeń.

Można to naprawić ponownie uruchamiając NVmakedevices.sh, a następnie wyedytować plik /etc/conf.d/rc i ustawić:

Listing 3.3: Edytowanie /etc/conf.d/rc

```
RC_DEVICE_TARBALL="yes"
```

To spowoduje zachowanie plików /dev/nvidia* nawet po restarcie komputera.

Uwaga: Można również spróbować zainstalować najnowsze pakiety nvidia-kernel oraz nvidia-glx z serii 8xxx. W chwili pisania tego artykułu oba są zamaskowane flagami ~arch. Te sterowniki niepolecają już na udev i hotplug, więc nie ma potrzeby uruchamiania NVmakedevices.sh. W zamiast sam sterownik X utworzy plik urządzenia /dev/nvidia* podczas startu X-ów. Jednakże to wymaga używania nowych wersji jądra (2.6.14 lub nowszych).

4. Konfiguracja dla ekspertów

Dokumentacja

Pakiet sterownika nvidia zawiera mnóstwo użytecznej dokumentacji, która jest instalowana do katalogu /usr/share/doc i może być odczytana przy pomocy następującego polecenia:

Listing 4.1: Przeglądanie dokumentacji NVIDIA

```
# less /usr/share/doc/nvidia-glx-*/README.txt.gz
```

Parametry modułów jądra

Moduł nvidia można uruchomić z jedną z kilku opcji, które pozwalają dostosować jego zachowanie do naszych potrzeb. Aby dodać lub usunąć te opcje należy wyedytować plik /etc/modules.d/nvidia. Po wprowadzeniu zmian należy uruchomić polecenie modules-update i przeładować moduł.

Listing 4.2: Dostosowywanie ustawień

```
(Otwieramy /etc/modules.d/nvidia ulubionym edytorem)
# nano -w /etc/modules.d/nvidia
(Aktualizujemy informacje o sterownikach)
# modules-update
(Usuwamy moduł z pamięci...)
# modprobe -r nvidia
(...i ładujemy go na nowo)
# modprobe nvidia
```

Zaawansowana konfiguracja X-ów

Nakładka GLX również posiada wiele opcji, które można dostosować do własnych potrzeb. Za ich pomocą można np. zmienić zachowanie wyjścia TV, wykrywanie częstotliwości monitora itp. Po raz kolejny dostępne opcje są opisane w szczegółowej dokumentacji.

Aby używać którejś z tych opcji należy dopisać ją w odpowiednim miejscu w pliku konfiguracyjnym serwera X (zwykle jest to /etc/X11/xorg.conf). Na przykład, aby wyłączyć wyświetlanie logo nvidii przy każdym starcie serwera X należy wpisać:

Listing 4.3: Dodatkowe opcje konfiguracyjne

```
Section "Device"
Identifier "nVidia Inc. GeForce2"
Driver      "nvidia"
Option      "NoLogo" "true"
VideoRam   65536
EndSection
```

3. Instalacja i konfiguracja kart ATI

1. Obsługa sprzętu

Większa część kart ATi jest obsługiwana przez [xorg-x11](#), przynajmniej jeśli chodzi o przyspieszenie grafiki 2D. Obsługa grafiki 3D jest możliwa dzięki [projektowi DRI](#), który już jest zawarty w xorg-x11, lub przez zamknięte sterowniki [dostarczane przez ATi](#).

GPU	Nazwa	Obsługiwane przez
Rage128	Rage128	xorg DRI
R100	Radeon7xxx, Radeon64	xorg DRI
R200, R250, R280	Radeon8500, Radeon9000, Radeon9200	xorg DRI, ATI DRI
R300, R400, R500	Radeon 9500 - x800	xorg 2D, ATI DRI

Mam kartę typu All-In-Wonder/Vivo. Czy funkcje multimedialne są obsługiwane?

Obsługę funkcji multimedialnych zapewnia [projekt GATOS](#), te sterowniki zostaną niedługo włączone do xorg-x11.

Obsługa związana z X11 jest niemal równie dobra na platformach ppc i alpha, ale nie ma możliwości używania zamkniętych sterowników ATi. Oznacza to, że niemożliwe jest skorzystanie z funkcji 3D kart r300 i nowszych. Posiadacze takich kart powinni skontaktować się z [ATi](#) i poprosić o wydanie otwartych sterowników.

Ważne: Na niektórych płytach głównych AMD64 należy wyłączyć obsługę K8 IOMMU przed włączeniem obsługi agpgart.

Posiadam laptopa. Czy moja karta "mobility" jest obsługiwana?

Powinna być, ale może wystąpić błąd konfiguracji związany z OEM PCI id, który występuje na tych chipach. W większości przypadków trzeba będzie samodzielnie napisać plik konfiguracyjny lub użyć xorgconfig.

2. Instalacja

Pakiety

- Ebuild xorg-x11 dostarcza implementacji X11.
- Dla jąder z serii 2.6.x moduły jądra i dri mogą być budowane razem z jądrem lub zostać zainstalowane poprzez ebuild x11-drm
- Dla jąder z serii 2.4.x trzeba użyć ebuildu x11-drm
- Sterowniki ATi dostarczane przez ebuild ati-drivers zawierają zarówno zamknięte sterowniki ATi jak i moduły jądra dla jąder serii 2.4.x i 2.6.x

Jeśli planujemy użycie wbudowanej w sterowniki ATi obsługi agpgart zamiast tej z jądra Linuksa, moduł agpgart i moduł specyficzny dla naszego chipu (w konfiguracji jądra) muszą być wyłączone.

Listing 2.1: Instalacja sterowników

```
Dla sterowników Rage128 instalowanych razem z X11
# VIDEO_CARDS="r128" emerge x11-drm
```

```
Dla kart Radeon
(R100, R200, R250, R280 ale jeszcze nie R300)
# VIDEO_CARDS="radeon" emerge x11-drm
By zainstalować zamknięte sterowniki ATI.
(tylko R200, R250, R280 i R300)
# emerge ati-drivers
```

```
By zainstalować X11, ale nie instalować
żadnych modułów jądra
# emerge xorg-x11
```

Konfiguracja

Sugerujemy użycie xorgcfg, xorgconfig lub bezpośrednie skorzystanie z opcji autokonfiguracji Xorg:

Listing 2.2: Automatyczny konfiguracja X

```
# X -configure
```

Więcej informacji o samym pliku konfiguracyjnym xorg.conf można zdobyć w opisie konfiguracji serwera X.

Uwaga: Użytkownicy ati-drivers mogą także użyć aticonfig.

Ważne: Użytkownicy PPC po zainstalowaniu ebuildu Xorgautoconfig mogą skorzystać z niezależnego konfiguratora o nazwie Xorgautoconfig.

Przelaczanie bibliotek OpenGL

Kiedy już zainstalujemy X-y, skonfigurujemy je i uruchomimy są one gotowe do używania bibliotek OpenGL od ATI:

Listing 2.3: Uruchamianie eselect

```
# eselect opengl set ati
```

4. Instalacja i konfiguracja KDE

1. Co to jest K Desktop Environment?

Projekt

Projekt KDE to projekt, który ma na celu tworzenie darmowego oprogramowania dedykowanego dla środowiska KDE - graficznego środowiska o otwartym kodzie, przeznaczonego dla stacji roboczych Linux oraz UNIX. Rozwijaniem projektu zajmuje się kilkuset inżynierów oprogramowania z całego świata, którzy zgrodzili się za darmo pracować dla projektu. Więcej informacji o KDE można uzyskać na stronach tego projektu.

Oprogramowanie

K Desktop Environment to proste w użyciu środowisko graficzne zbudowane na dobrze przemyślanym szkielecie programowym. Pozwala on na współdziałanie między aplikacjami, wykonywanie operacji na zasadzie "przeciagnij i upuść", itp. Oprócz obowiązkowych komponentów wymaganych przez środowisko, KDE zapewnia również odpowiednie aplikacje do setek zastosowań: zarządzania plikami, przeglądania stron WWW, prac biurowych, czy obsługi poczty elektronicznej. Wszystko to wchodzi w skład projektu KDE.

Środowisko KDE jest dostępne w ponad 70 językach i jest używane przez ogromne rzesze użytkowników. Dla zainteresowanych, dostępnych jest wiele zrzutów ekranów. Aby uzyskać więcej informacji na temat KDE, można przeczytać artykuł [Co to jest KDE?](#) (w języku angielskim) na stronie domowej KDE.

Społeczność

Istnieje wiele stron społeczności KDE. Na stronie [KDEnews.org](#) można znaleźć najnowsze ogólne informacje na temat KDE. [KDEdevelopers.org](#) z kolei przeznaczone jest dla osób rozwijających KDE, zaś [KDE-forum](#) jest przeznaczone do komunikacji pomiędzy użytkownikami.Więcej odnośników można znaleźć na stronie rodziny KDE.

2. Instalowanie KDE

Co jest potrzebne?

Jeśli jesteśmy zainteresowani zainstalowaniem KDE (lub wsparciem projektu KDE), musimy upewnić się, że we flagach USE mamy ustawioną flagę kde oraz qt3 lub qt4 (lub obie naraz). Zainteresowani powinni wiedzieć, że Qt jest biblioteką graficzną niezbędną dla poprawnego działania KDE. qt3 odpowiada za wsparcie wersji 3.x, natomiast qt4 odpowiada za wsparcie dla nowszej wersji 4.x.

KDE potrafi również samodzielnie montować urządzenia. Należy dodać hal do flag USE, jeśli chcemy zapewnić obsługę automatycznego montowania urządzeń, tak jak to opisano w części [Montowanie urządzeń w KDE](#).

Jeśli nie będzie używane [aRts](#) do obsługi multimedialnych, należy wyłączyć flagę USE arts (jest ona domyślnie włączona).

Uwaga: W wydaniu Gentoo 2006.1 pojawił się szereg nowych profili, w tym podprofil o nazwie desktop. Użytkownicy KDE mogą chcieć się na niego przełączyć, jeśli jest dostępny dla ich architektury, ponieważ włącza on domyślnie serię flag USE, które w innym wypadku musieliby włączać ręcznie. Opis zmiany profilu znajduje się w tekście [Aktualizowanie Gentoo](#).

Instalowanie KDE przy pomocy monolitycznych pakietów

Projekt KDE opublikował nową wersję środowiska, składającą się z 16 dużych pakietów. Każdy z nich zawiera wiele aplikacji (dlatego nazywane są monolitycznymi). Musimy zdecydować, które pakiety zainstalujemy.

Możemy przejrzeć listę wszystkich pakietów jakie mogą być zainstalowane:

Listing 2.1: Przedstawienie wszystkich pakietów które zostaną zainstalowane z KDE

```
# emerge --pretend kde | less
```

Jeśli nie muszą być instalowane wszystkie pakiety, można zainstalować poszczególne osobno. Na pewno należy zainstalować pakiet kdebase, ponieważ zawiera on trzon KDE oraz wszystkie składniki potrzebne do uruchomienia czegokolwiek z KDE. Poniżej znajduje się tabela z dostępnymi pakietami które można zainstalować.

Pakiet	Opis
kdeaccessibility	Programy przeznaczone dla osób niepełnosprawnych, tworzone w ramach projektu KDE Accessibility Project
kdeadmin	Programy administracyjne KDE, takie jak KCron , KUser (zarządzanie użytkownikami) oraz KDat (zarządzanie kopiami bezpieczeństwa)
kdeartwork	Różne aplikacje związane ze sztuką komputerową, takie jak wygaszacz ekranu czy tematy pulpitu. Na stronie artist.kde.org można znaleźć więcej informacji na ich temat.
kdeedu	Aplikacje edukacyjne KDE przeznaczone dla osób w wieku od 3 do 18 lat. Na stronie KDE Edu Project można znaleźć więcej informacji.
kdegames	Różne gry stworzone dla KDE.Więcej informacji można znaleźć pod adresem KDE Games Center.
kdegraphics	Oprogramowanie związane z grafiką, przeznaczone dla KDE, w tym KSsnapshot (oprogramowanie do tworzenia zrzutów ekranu), KolourPaint (prosty edytor grafiki), Kpdf (przeglądarka plików PDF), KIconEdit (edytor ikon) oraz KPovModeler (oprogramowanie do modelowania w trzech wymiarach).
kdemultimedia	Oprogramowanie związane z multimediami, włączając wsparcie dla aplikacji obsługujących dyski CD, MP3, DVD, dźwiękowych i wideo. Więcej informacji znajduje się na stronie projektu KDE Multimedia Project.
kdenetwork	Aplikacje związane z pracą w sieci, jak Kopete (wieloprotokołowy komunikator), kppp (Dial-In) i KSirc (klient IRC). Należy pamiętać, że konqueror (menedżer plików i przeglądarka) jest częścią pakietu kdebase
kdepim	Narzędzia do zarządzania informacjami osobistymi, takie jak KOrganizer (dziennik), KAddressbook (książka adresowa), Kontakt (praca grupowa) oraz KMail (poczta elektroniczna). Więcej informacji o nich znajdziemy na stronie projektu KDE PIM Project.
kdesdk	Narzędzia do rozwijania kodu, włączając KBabel (narzędzie do tłumaczeń), KBugBuster (interfejs użytkownika do śledzenia dziur w KDE) oraz Kompare (interfejs do przedstawiania różnic w plikach).
kdetoys	Różne zabawki które przynoszą rozrywkę w trakcie oczekiwania na przybycie pizzy. Znajdują się tu takie aplenty jak eyesapplet oraz fifteenapplet, ale również takie narzędzia jak amor, które nie robią nic oprócz zjadania zasobów :)
kdeutils	Graficzne narzędzia systemowe takie jak kcalc (kalkulator), kdesh (terminal SSH), kfloppy (czynności związane ze stacjami dyskietek), itp.
kde-i18n	Internacjonalne elementy KDE. Pakiet zawiera m.in. przetłumaczoną dokumentację. Więcej informacji uzyskać można na stronie projektu KDE i18n.

Dla przykładu, aby zainstalować KDE jedynie z aplikacjami związanymi z siecią oraz czynnościami administracyjnymi wykonamy polecenie:

Listing 2.2: Przykładowa instalacja pojedynczych komponentów KDE

```
# emerge kdebase kdenetwork kdeadmin
```

W przypadku wątpliwości: komplikacja KDE troszkę trwa.

Instalacja KDE przy użyciu rozdzielonych pakietów

Jeśli chcemy mieć jeszcze większą kontrolę, nad tym jakie elementy KDE zostaną zainstalować, mamy możliwość wyboru pojedynczych programów, które będziemy wykorzystywać. Więcej informacji na temat ebuildów poszczególnych programów KDE można uzyskać w dokumencie Rozdzielone ebuildy KDE.

Wybór co zainstalować, a co nie, w przypadku rozdzielonych ebuildów jest znacznie bardziej skomplikowany niż gdy korzystamy z monolitycznych pakietów. Dlatego też Gentoo dostarcza meta pakietu, który zawiera pewną liczbę programów KDE:

- W przypadku zapotrzebowania na pełną instalację KDE, należy zainstalować kde-meta. Ten pakiet zainstaluje wszystkie aplikacje KDE jako zależności.
- Podstawową instalację KDE można osiągnąć instalując kdebase-startkde. W dowolnym momencie można instalować kolejne aplikacje KDE.

Powyższe dwie opcje to skrajne możliwości. Najlepszym rozwiązaniem byłoby zastosowanie bezpiecznej mieszanki obu. Aby ułatwić proces podejmowania decyzji, poniższa tabela przedstawia krótki, niekompletny przegląd niektórych dostępnych pakietów KDE.

Te pakiety nie są częścią instalacji kdebase-startkde.

Nazwa pakietu	Opis
akregator	Aplikacja to łatwego zarządzania i przeglądania kanałów RSS.
juk	Obsługujący playlisty odtwarzacz multimedialny, inspirowany iTunes firmy Apple.
kate	KDE Advanced Text Editor (KATE) to edytor pozwalający na edytowanie wielu dokumentów równocześnie z kolorowaniem składni, zwijaniem kodu oraz wieloma innymi możliwościami.
kcontrol	Centrum Sterowania KDE.
kicker	kicker to powszechnie znany panel do uruchamiania aplikacji. Zawiera Menu K i umożliwia uruchamianie apletów takich jak pager, zasobnik systemowy, zegar, itp.
kmail	Pocztę można efektywnie zorganizować dzięki KMail.
knetattach	Dzięki KNetAttach (również znane jako Network Folder Wizard), można w prosty sposób dodać zasób sieciowy do pulpitu KDE.
knodle	KNode to posiadająca wiele możliwości przeglądarka newsów.
konqueror	Konqueror to w pełni funkcjonalny menedżer plików oraz dobra przeglądarka stron WWW.
konsole	Konsole to emulator terminala przeznaczony dla KDE.
kontact	Kontact to menedżer informacji osobistych przeznaczony dla KDE, pozwalający zarządzać łatwiej kontaktami, oraz zorganizować wspólną pracę aby przebiegała szybciej i sprawniej.
kopete	Kopete to komunikator przeznaczony dla KDE wspierający wszystkie znane protokole różnych komunikatorów.
korganizer	Korganizer to kalendarz i aplikacja zarządzająca czasem dla KDE.
kpdf	Przy pomocy KPDF można przeglądać i pracować z plikami PDF. Program posiada unikalne cechy, które znacznie ułatwiają przeglądania plików.
kscd	kscd to graficzny odtwarzacz płyt CD dla KDE.
ksnapshot	Przy pomocy kssnapshot można robić zrzuty ekranu.
kuickshow	kuickshow umożliwia wyświetlanie wielu rodzajów plików graficznych.

Jest to dopiero wierzchołek góry lodowej. Aby uzyskać informacje o wszystkich aplikacjach KDE, wystarczy zajrzeć do kategorii kde-base . Przeznaczenie poszczególnych programów powinno być dostępne w opisie.

Aby podejrzeć co zainstaluje emerge, należy użyć emerge -p z pomocą less, w innym przypadku może być kłopot z zobaczeniem wszystkich paczek.

Listing 2.3: Podglądanie instalacji kde

```
(Należy wstawić wybrane pakiety)
# emerge -p kdebase-startkde kicker | less
```

Jeśli rezultat jest poprawny, wystarczy usunąć -p. Kompilacja zapewne troszeczkę potrwa. KDE jest olbrzymim środowiskiem, więc nie ma się co dziwić, że nie zostanie skompilowane w ciągu kilku minut.

Dodatkowe aplikacje KDE

Aplikacje KDE nie ograniczają się tylko do tych, które dołączane są do oficjalnych wydań tego środowiska. Istnieje wiele programów, które wykorzystują szkielet KDE i jego biblioteki. Poniżej znajduje się lista tylko tych najbardziej popularnych:

Nazwa ebuilda	Opis
koffice	KOffice to obszerny pakiet biurowy, w którego skład wchodzą aplikacje przeznaczone do edycji tekstu (KWord), pracy w arkuszu kalkulacyjnym (KSpread), tworzenia prezentacji (KPresenter), edycji grafiki (Krita), zarządzania bazami danych (Kexi) i wiele innych. Podobnie jak KDE instalowane przez ebuildy kde lub kde-meta, KOffice może zostać zainstalowany przy użyciu monolitycznego pakietu (koffice) lub zestawu pojedynczych pakietów (koffice-meta).
amarok	amaroK to posiadający wiele możliwości odtwarzacz muzyczny dla systemów Unix/Linux
k3b	K3B stanowi kompletne narzędzie do nagrywania płyt CD/DVD z obsługą płyt audio. Nagrywanie płyt przy jego użyciu jest bardzo proste.
kaffeine	Kaffeine to rozbudowany odtwarzacz multimedialny dla KDE.

Pierwsze wrażenia

Spójrzmy teraz na rezultat. Należy przestrzegać tego czego mama od zawsze nauczała o pracy na koncie roota. Dlatego też należałoby posłuchać rad mamy i uruchomić KDE na koncie zwykłego użytkownika. Trzeba skonfigurować sesję aby system uruchamiał KDE po wpisaniu startx w linii poleceń. Wystarczy wpisać exec startkde w ~/.xinitrc (więcej informacji w akapicie Używanie startx w dokumencie Konfiguracja serwera X):

Listing 2.4: Konfigurowanie lokalnej sesji

```
$ echo "exec startkde" > ~/.xinitrc
```

Teraz trzeba wpisać startx w linii poleceń, aby uruchomić środowisko graficzne.

Listing 2.5: Uruchamianie KDE

```
$ startx
```

Na powitanie pojawi się aplikacja zwana KPersonalizer. Gratulacje, teraz trzeba tylko skonfigurować KDE...

3. Konfiguracja KDE

KPersonalizer

KPersonalizer to aplikacja, która dokonuje konfiguracji KDE. Jest to bardzo przydatny kreator, który pozwala w szybki sposób dostosować KDE do własnych potrzeb. Przy pierwszym uruchomieniu KDE, KPersonalizer jest uruchamiane automatycznie.

Pierwsze dane jakie należy podać to preferowany kraj oraz język. Jako że jeszcze nie zostały zainstalowane pakiety lokalizacyjne, dostępne języki będą bardzo ograniczone -- zapewne do wyboru będzie jedynie język angielski. Nie należy się tym przejmować, gdyż język będzie można później zmienić.

Kolejny wybór jaki jest przedstawiany to sposób zachowania się systemu. W tym zachowanie się okien, wybór myszą, itd. Przy wybieraniu danego zachowania, jego opis jest przedstawiany, aby pomóc przy dokonywaniu wyboru. W razie niepewności, nie ma co panikować -- można zmienić to zachowanie w dowolnym momencie.

W kolejnym kroku, KPersonalizer pyta o ilość graficznych urozmaiceń które powinny być włączone. Im więcej, tym ładniej będzie wyglądać KDE, jednak procesor będzie bardziej obciążony. Jednakże nie jest aż tak źle - na systemie z procesorem 600MHz i 128Mb pamięci, włączenie maksymalnej ilości urozmaiceń graficznych nadal pozwala na w miarę wygodną pracę.

Ostatecznie, KDE pyta o styl jaki ma być używany. Styl opisuje dekorację okna, temat, wygląd przycisków itd. Małe okienko pozwala zorientować się w ich wyglądzie. Czy wspominaliśmy już, że KDE jest bardzo konfigurowalne?

Teraz można usiąść i się cieszyć się -- KDE się uruchomi, a na ekran zawita ładne, uporządkowane i funkcjonalne środowisko graficzne.

Instalowanie innych języków

Ta sekcja przeznaczona jest dla osób, których język macierzowy jest inny niż angielski, lub które chcą po prostu pracować w KDE w innym języku. Zostanie tu opisane jak zainstalować inne języki.

Paczki językowe znajdują się w paczce kde-i18n. Aby zainstalować wybrany pakiet lokalizacyjny, należy ustawić zmienną LINGUAS, aby wskazywała na odpowiedni(e) język(i). Rozsądne jest aby umieścić ją w /etc/make.conf tak, aby aktualizacja systemu nie usuwała pożądanych pakietów językowych.

Listing 3.1: Ustawianie LINGUAS w /etc/make.conf

```
# nano -w /etc/make.conf
(Jako przykład, zostaną zainstalowane pakiety dla języka polskiego (pl)
 oraz francuskiego (fr))
LINGUAS="pl fr"
```

Teraz trzeba wpisać emerge kde-i18n, aby zainstalować odpowiednie pakiety. Po instalacji, wystarczy ponownie uruchomić KDE i przejść do Centrum Sterowania KDE (K-menu > Control Center). Jest to miejsce w którym można kontrolować niemalże każdą część KDE. Jest też ono znacznie bardziej rozbudowane niż KPersonalizer.

Aby zmienić język, należy wejść do Regional & Accessibility, Country/Region & Languages. Tam należy dodać wybrane języki. Aby ujrzeć zlokalizowane KDE, wystarczy uruchomić go ponownie.

Graficzny ekran logowania

Aby używać kdm jako graficznego menadżera logowania (co oznacza, że nie trzeba będzie logować się do terminala i wpisywać startx za każdym razem), trzeba zmienić jeden plik konfiguracyjny i ustawić system, aby uruchamiał się w trybie graficznym tuż po wystartowaniu, zgodnie z opisem znajdującym się poniżej.

W pliku /etc/rc.conf, trzeba ustawić w zmiennej DISPLAYMANAGER wartość kdm.

Listing 3.2: Ustawianie zmiennej DISPLAYMANAGER w /etc/rc.conf

```
# nano -w /etc/rc.conf
(Trzeba wyedytować poniższą zmienną)
DISPLAYMANAGER="kdm"
```

Na zakończenie należy dodać xdm do domyślnego poziomu uruchamiania:

Listing 3.3: Dodawanie xdm do domyślnego poziom uruchamiania

```
# rc-update add xdm default
```

Po ponownym rozruchu systemu, KDM zostanie użyte jako graficzny ekran logowania.

KDM wyświetli listę dostępnych do uruchomienia sesji, jedną z nich oczywiście będzie KDE, pozostałe pozycje będą zależne od zainstalowanego w systemie oprogramowania. Sesje dla wszystkich menedżerów okien znajdują się w katalogu /usr/share/xsessions/. Jeśli korzysta się z KDM nie trzeba zmieniać pliku ~/.xinitrc.

Montowanie urządzeń w KDE

KDE daje możliwość montowania urządzeń takich jak napęd CD-ROM czy pamięć USB poprzez pojedyncze kliknięcie w graficznym interfejsie. Aby skorzystać z tej funkcjonalności konieczne jest skompilowanie KDE z flagą USE hal i zainstalowanie programów dbus i hal. Należy również dodać dbus i hal do domyślnego poziomu uruchamiania, a użytkownicy powinni być członkami grupy plugdev.

Listing 3.4: Ustawienia dla montowania urządzeń

```
# emerge --noreplace dbus hal
# rc-update add dbus default
# rc-update add hald default
Add <user> to the plugdev group
# gpasswd -a <user> plugdev
```

Możliwe jest także montowanie urządzeń bez jakiegokolwiek interakcji ze strony użytkownika. Jeśli chcemy skorzystać z tej możliwości musimy zainstalować program ivman i dodać go do domyślnego poziomu uruchamiania.

Listing 3.5: Automatyczne montowanie

```
# emerge --noreplace ivman
# rc-update add ivman default
```

4. Zarządzanie instalacjami KDE

Wielokrotne instalacje

Jedną ze specyficznych cech zarządzania instalacjami KDE w Gentoo jest to, że kiedy nowa seria KDE zostaje opublikowana (jak na przykład seria 3.5.x, zastępująca serię 3.4.x), zostanie ona zainstalowana obok starej i jej nie nadpisze. Przykładowo, jeśli posiadamy KDE 3.4 i zemerujemy KDE 3.5, to będziemy posiadali w systemie dwie wersje, jedną zainstalowaną w katalogu /usr/kde/3.4/ i drugą w /usr/kde/3.5/.

Należy zwrócić uwagę na fakt, że wersje KDE z różnych serii posiadają oddzielne katalogi z konfiguracją w katalogu domowym użytkownika. KDE 3.4 wczytuje swoje ustawienia z katalogu /home/<user>/.kde3.4. Gdy po raz pierwszy uruchomimy KDE 3.5, utworzony zostanie katalog /home/<user>/.kde3.5, do którego przeniesiona zostanie konfiguracja wersji 3.4. Pliki konfiguracyjne dwóch serii mogą być modyfikowane niezależnie.

Kolejną sprawą, o której należy pamiętać w czasie aktualizacji KDE, jest możliwość wystąpienia problemów z dodatkowymi aplikacjami KDE, które wcześniej zainstalowaliśmy (jak koffice, amarok lub k3b). Konieczne jest przekompilowanie takich programów w obecności nowej wersji KDE. Gdy tylko zaczniemy używać nowej serii KDE, musimy powtórnie zemerować te aplikacje, aby zostały one powiązane z nowymi bibliotekami.

Odinstalowywanie starej wersji

Posiadanie wielu wersji KDE zainstalowanych jednocześnie wywołuje problem, w jaki sposób usunąć starsze wersje, gdy zdecydujemy, że nie są one nam już potrzebne. Niestety portage nie daje możliwości odinstalowania pakietu i wszystkich jego zależności przy pomocy pojedynczej komendy, więc jeśli wykonamy polecenie emerge --unmerge kde, w rzeczywistości nie usuniemy pakietów KDE.

Aby usunąć jakąś wersję KDE (przykładowo 3.4), poszczególne pakiety muszą zostać usunięte.

Listing 4.1: Usuwanie pakietów KDE 3.4

```
# emerge --unmerge =arts-3.4* =kdelibs-3.4* =kdebase-3.4* ...
```

Oczywiście jest to sposób skrajnie niewygodny, jeśli zainstalowaliśmy wiele programów KDE. Powyższą operację można zautomatyzować na wiele sposobów, poniższy jest tylko przykładem.

Na początku wygenerujemy listę wszystkich pakietów, jakie chcemy usunąć. Posłużymy się narzędziem equery, który jest częścią pakietu app-portage/gentoolkit:

Listing 4.2: Generowanie listy pakietów do usunięcia

```
(Wszystkie zainstalowane pakiety KDE)
# equery list kde-base/
(Zainstalowane pakiety KDE 3.4)
# equery list kde-base/ | grep 3\.4
```

Należy dwukrotnie sprawdzić, czy lista pakietów zawiera programy, które powinniśmy usunąć z systemu. Jeśli lista jest poprawna, możemy przekazać ją jako argument dla komendy emerge --unmerge.

Listing 4.3: Usuwanie wybranych pakietów

```
# equery list kde-base/ | grep 3\.4 | xargs emerge --unmerge --pretend
```

Sprawdźmy jeszcze raz listę programów i jeśli jest ona poprawna powtórzmy powyższą komendę bez parametru --pretend, aby rozpoczęć proces odinstalowywania.

Gdy ten proces zakończy się, katalog /usr/kde/3.4/ powinien zawierać tylko kilka plików (głównie pliki konfiguracyjne - polityka portage zakłada, aby nigdy nie ingerować w konfigurację). Jeśli chcemy, możemy bezpiecznie usunąć katalog /usr/kde/3.4/ razem z jego zawartością, aby pozbyć się pozostałości KDE 3.4.

5. Często zadawane pytania

KDE jest niezwykle wolne podczas startu

Trzeba upewnić się, że plik /etc/hosts jest prawidłowy:

- W przypadku statycznego numeru IP, trzeba upewnić się, że FQDN oraz nazwa hosta są odpowiednio wpisane, np. 192.168.0.10 tux.mydomain tux
- W przypadku adresu przydzielanego dynamicznie lub gdy nie ma w ogóle dodatkowych interfejsów, należy dodać nazwę hosta w linijce z adresem localhost, np. 127.0.0.1 localhost tux

Należy upewnić się, że dla dysków uruchomione jest DMA:

Listing 5.1: Sprawdzanie ustawień DMA

```
# hdparm /dev/hda
(...)
using_dma      = 1 (on)
(...)
```

5. Rozdzielone ebuildy w KDE

1. Rozdzielone ebuildy KDE.

Czym one są?

Do stycznia 2005 roku wszystkie te ebuildy w Portage były monolityczne. Było ich tylko 15 (kdebase, kdenetwork, ...) i każdy instalował wiele aplikacji, które w zasadzie nie były zależne od siebie. Nie była to do końca optymalna sytuacja i z pewnością niezbyt zgodna z założeniami Gentoo, jednak była tolerowana przez długi okres czasu.

Nowe rozdzielone pakiety(dla konquerora, kmaila, ...) poprawiły tą sytuację dostarczając odseparowane ebuildy dla każdej poszczególnej aplikacji KDE. To dało nam całkowitą liczbę 330 nowych ebuildów w kategorii kde-base.

Ciągle udostępniamy monolityczne ebuildy dla KDE 3.4 i 3.5. Jednak to te rozdzielone są nowym standardem i nie przewiduje się tworzenia tych monolitycznych dla KDE w wersji 4.0.

Wreszcie, należało by wspomnieć, że istnieją także rozdzielone ebuildy dla Koffice. Instalują one osobno takie programy jak kword, kugar itp.

Jak instalować rozdzielone ebuildy

W momencie pisania tego tekstu ostatnim stabilnym wydaniem KDE było 3.4.3, natomiast ostatnim niestabilnym (package.mask) wydanie 3.5.0_beta2. Rozdzielone i monolityczne ebuildy są dostępne w drzewie Portage.

- Aby zemergować poszczególne pakiety, takie jak np. kmail, po prostu wykonujemy emerge kmail
- Aby zemergować podstawowe środowisko kde, umożliwiające zalogowanie się do minimalnej sesji KDE, wpisujemy emerge kdebase-startkde
- Aby uzyskać dokładny odpowiednik jednego z monolitycznych pakietów - na przykład, aby zainstalować wszystkie aplikacje znajdujące się w kdebase za pomocą podzielonych ebuildów - można wykonać emerge kdebase-meta (lub kdepim-meta dla kdepim, itd.) Aby zainstalować absolutnie wszystkie rozdzielone ebuildy KDE używamy emerge kde-meta.

Jak przejść z monolitycznego KDE na rozdzielone ebuildy?

Kiedy jest zainstalowane KDE 3.3.x, można po prostu wpisać polecenie emerge kde-meta, aby zainstalować rozdzielone KDE 3.4.x bez naruszania obecnej instalacji. To samo dotyczy się wersji 3.5.x.

Jeśli zainstalowane jest monolityczne KDE 3.4.x, trzeba najpierw je usunąć, a dopiero potem przystąpić do instalacji wersji podzielonej. Można tego dokonywać tylko na wybranych pakietach, nie trzeba od razu usuwać całości.

Nie należy obawiać się, że przejście na rozdzielone ebuildy może cokolwiek zepsuć. Portage posiada system blokad, które zapewniają, że jeśli coś da się bezpośrednio zainstalować, to będzie to działało prawidłowo.

Zalety rozdzielonych ebuildów.

Oto krótka lista korzyści z przejścia na rozdzielone ebuildy:

- Większość programów KDE nie zmienia się pomiędzy drugorzędnymi wydaniami KDE. Na przykład, aktualizacja KDE z wersji 3.3.1 na 3.3.2 zmienia mniej niż 100 aplikacji z 320. Podzielone paczki, umożliwiają tworzenie nowych ebuildów tylko dla paczek, które zostały zmienione, oszczędzając (w tym przykładzie) więcej niż dwie trzecie czasu komplikacji na aktualizację.
- Poprawki zazwyczaj dotyczą konkretnych aplikacji. Dzięki nowej filozofii mogą być testowane, zatwierdzane i oddawane szybciej, a więc deweloperzy będą mieli mniej do zrobienia. Poza tym, jak już wyżej napisałem, zwykły użytkownik będzie zużywał znacznie mniej czasu na aktualizację, co jest to szczególnie ważne przy aktualizacjach związanych z bezpieczeństwem.
- Użytkownicy innych środowisk graficznych i lżejszych menadżerów okien mogą zemergować kilka aplikacji KDE, jeśli zechcą, bez ogromnych zależności, które powodowały stare ebuildy takie jak kdebase czy kdepim.
- Użytkownicy mogą wybrać teraz aplikacje jakie mają zainstalowane. Powody mogą być różne:
 - Zależy im na czasie komplikacji. emerge kdebase kdepim kdenetwork trwa strasznie długo, a tak naprawdę potrzebne im są jedynie konqueror, kmail i kopete.
 - Zależy im na niezaśmiecaniu dysku. Każda nieużywana aplikacja sprawia, że cenne megabajty się marnują. Dysk z większą ilością wolnego miejsca też wtedy oddycha swobodniej; jest szybszym i szczęśliwszym dyskiem.
 - Troszczą się o bezpieczeństwo systemu. Każde zainstalowane oprogramowanie jest potencjalnym celem ataku i dlatego nie należy instalować niczego co nie jest potrzebne.
 - Dokładnie poznali filozofię Gentoo i nie mogą znieść tak wielu programów ściśniętych w jeden pakiet? My też nie możemy.
 - Ostatecznie, należy zaznaczyć, że rozdzielone ebuildy zapewniają znacznie większą elastyczność w definiowaniu dla nich flag USE.

Współpraca rozdzielonych i monolitycznych ebuildów.

Rozdzielone i monolityczne ebuildy mogą swobodnie razem współpracować. Jedynym ograniczeniem jest to, że ebuild monolityczny nie może być jednocześnie zainstalowany razem z rozdzielonym pochodzący od niego. W ebuildach KDE istnieją mechanizmy blokowania nie zezwalające na to, zatem można robić wszystko na co tylko emerge pozwoli.

Jednak zwykle nie ma powodu, aby używać takich mieszanych konfiguracji. W rzeczywistości, za wyjątkiem specjalnych

przypadków jak na przykład bardzo wolne komputery (mips), należy używać rozdzielonych ebuildów do wszystkich swoich potrzeb.

Rozdzielone ebuildy są także tymi domylnymi. Oznacza to, że kiedy jakieś inne ebuildy zależą od aplikacji KDE to będą chciały instalować właśnie te rozdzielone ebuildy. Jednakże monolityczne ebuildy także spełnią te zależności, więc można zemergować monolityczny ebuild ręcznie i wtedy dopiero ebuild który od niego zależał.

2. Problemy z wydajnością.

Dlaczego rozdzielone ebuildy są takie powolne?

Mówiliśmy już dawno, że rozdzielone ebuildy instalują się dłużej niż te monolityczne, ponieważ dodatkowo dla każdej aplikacji musi zostać uruchomione rozpakowywanie i uruchamianie skryptu konfiguracyjnego. Kompletne emerge kde-meta może zabrać około 20-30% więcej czasu niż klasyczne emerge kde, które i tak zajmowało go już mnóstwo.

Co więcej, teraz każdy z rozdzielonych ebuildów uruchamia make -f admin/Makefile.cvs (to oznacza uruchomienie autoconf, automake, itd. oraz kilka powiązanych specyficznych dla KDE skryptów). To powoduje dodatkowe narzuty czasowe w przybliżeniu równe tym spowodowanym przez skrypt configure.

Ostatecznie, rozdzielone ebuildy potrzebują wypakować specyficzne pliki z dużych archiwów. Jest to rozwiązanie wolniejsze od rozpakowywania małych dedykowanych archiwów. Jednak tworzenie takich małych archiwów dla systemów, na których KDE 3.x jest budowane przy pomocy autotools jest trudnym zadaniem.

Warto jeszcze raz podkreślić, że z rozdzielonymi ebuildami czas komplikacji przy aktualizowaniu KDE może zostać znaczaco skrócony poprzez aktualizacje tylko tych aplikacji KDE, które naprawdę się zmieniły. Korzyści jakie dają takie pojedyńcze aktualizacje aplikacji zwykle wynagradzają z nawiązką czas stracony podczas pierwszej instalacji KDE z nowych ebuildów.

W końcu instalacja całego KDE ma sens tylko jeżeli chcemy zobaczyć jakie aplikacje zawiera KDE lub jeśli tworzymy środowisko dla wielu użytkowników. Jednak większość ludzi używa tylko części z ponad 300 dostępnych aplikacji KDE. Każdy kto naprawdę troszczy się o czas komplikacji, np. właściciele starszych komputerów, mogą zyskać więcej czasu poprzez selektywną instalację programów, niż straciliby w związku z dodatkowymi nakładami czasowymi o których była mowa powyżej.

W jaki sposób rozdzielone ebuildy mogą być szybsze?

Większość lub nawet wszystkie problemy z wydajnością rozdzielonych ebuildów sprowadzają się do autotools - autoconf, automake i innych narzędzi, które zarządzają procesem budowania ./configure;make;make install KDE 3.x.

W KDE 4 (o ile nasze informacje są prawdziwe) zostanie zaadoptowany kompletnie nowy system komplikacji, który pośród wszystkich innych rzeczy, zdecydowanie zredukuje czas, w którym odpowiednik make -f admin/Makefile.common; ./configure zostanie wykonany. Mamy również nadzieję, że przyczyni się to do znacznie łatwiejszego tworzenia małych archiwów dla każdego rozdzielonego ebuildu poprzez obniżenie czasu generowania odpowiednika skryptów konfiguracyjnych (jeśli takie będą).

Poprzednio, confcache rozważane było jako sposób na obniżenie kosztów, które były ponoszone na wielokrotnie uruchamiane skrypty konfiguracyjne generowane przez autoconf. Confcache jest metodą przechowywania wyników testów konfiguracyjnych. Jednak confcache nie jest jeszcze zaimplementowany w stabilnej (2.0) wersji drzewa Portage. Nawet, jeżeli w przyszłości zostanie takowy zaimplementowany, może zostać wprowadzony za późno, aby używać go w ebuildach KDE. W takim wypadku będziemy czekać do wydania wersji 4 KDE.

3. Rozdzielne ebuildy - często zadawane pytania

Dlaczego niektóre rozdzielne pakiety nie posiadają nowszych wersji ebuildów?

Jak wyjaśniliśmy wcześniej nie wszystkie aplikacje są aktualizowane pomiędzy ważnymi wydaniami KDE, zatem nie wszystkie aplikacje otrzymują nowe wersje ebuildów. Dla przykładu, libkdenetwork nie został zaktualizowany w wersji 3.5.0_beta2, dlatego ostatnim ebuildem dostępnym z tym wydaniem był 3.5_beta1.

Jest to spowodowane wyłącznie chcią redukcji czasu potrebnego na komplikację podczas aktualizacji. Jeżeli stworzylibyśmy ebuilda libkdenetwork-3.5.0_beta2, zainstalowałby on dokładnie te same pliki jak ebuild 3.5_beta1. Dodatkowo aktualizowanych jest wiele zależności tak, aby wszystko działało poprawnie (np. aby żaden z ebuildów nie posiadał w zależnościach libkdenetwork-3.5.0_beta2).

Czy nie możemy zrobić teraz tego samego z wykorzystaniem DO_NOT_COMPILE?

DO_NOT_COMPILE jest zmienną środowiskową wewnętrznego systemu budowania KDE. Umożliwia selektywne wyłączenie podkatalogów z komplikacji. Niektórzy ludzie używali jej do komplikacji tylko części monolitycznego ebuildu KDE. Na przykład, uruchomienie polecenia DO_NOT_COMPILE=konqueror emerge kdebase zainstalowałoby kdebase bez konquerora.

Jednak DO_NOT_COMPILE nigdy nie było w założeniach narzędziem mającym ingerować w operacje menadżera automatycznego budowania pakietów. To po prostu nie działa, może nawet zepsuć system, a poza tym nie było nigdy wspierane. Namawiamy wszystkich, żeby wystrzegali się używania tego narzędzia.

A oto kawałek z listy problemów związanych z DO_NOT_COMPILE:

- Kompletnie psuje system śledzenia zależności zaimplementowany w Portage. Portage nie wie nic o DO_NOT_COMPILE i myśli że cały monolityczny pakiet został skompilowany i zainstalowany, więc uważa, że dana zależność musi być spełniona. Może to spowodować, że inne programy nie będą się chciały zbudować albo po prostu się nie uruchomią.
- Wymusza na użytkowniku konieczność znajomości nazw i znaczenia wszystkich istniejących podkatalogów z modułów KDE. Bardzo niewielu użytkowników poza deweloperami KDE, posiada o tym wiedzę, a więc mało kto jest w stanie odpowiednio używać DO_NOT_COMPILE.
- Poszczególne moduły w podkatalogach KDE mogą być powiązane między sobą zależnościami, mogą wymagać określonego porządku budowania lub obecności innego katalogu nawet jeżeli nie ma być on instalowany. Włożyliśmy dużo pracy w rozdzielone ebuildy tak, aby działały poprawnie pod tym względem. DO_NOT_COMPILE nie jest nawet w części tak dobrym narzędziem, żeby potrafiło uzyskać takie rezultaty, nawet z odpowiednią wiedzą ze strony użytkownika. Jedyną rzeczą jaką można z nim zrobić jest wyłączenie kilku aplikacji z komplikacji. Jest praktycznie niemożliwym, aby z pomocą tego narzędzia zainstalować tylko kilka aplikacji z modułu takiego jak kdebase czy kdepim.
- Jeśli zainstalowałem powiedzmy kmail wczoraj i dzisiaj chciałbym zainstalować korn używając DO_NOT_COMPILE, pociągnie to za sobą ponowną rekompilację kmail. Oznacza to, że DO_NOT_COMPILE jest zawsze dużo wolniejsze od rozdzielonych ebuildów.
- DO_NOT_COMPILE nie może zostać użyte do budowanie prekomplikowanych paczek (takich jak GRP) zawierających pojedyncze aplikacje KDE.

Czy nie powoduje to zbyt wielkiego obciążenia opiekunów KDE w Gentoo?

Co ciekawe, to pytanie było zadawane bardzo często. Jest mi bardzo miło, że użytkownicy tak dbają o nas, opiekunów ebuildów. Korzystając ze sposobności chcę zapewnić, że zajęliśmy się rozdzielonymi ebuildami KDE z własnej, nieprzymuszonej woli i że nie ma szans żeby ktoś nas od tego odwrócił. :-)

Powiniensem jeszcze wspomnieć, że opiekunowie innych architektur rzeczywiście narzekali, że będą musieli włożyć więcej wysiłku w testowanie i oznaczanie statusu mnóstwa ebuildów. Pracujemy nad rozwiązaniem tego problemu i właśnie to jest głównym powodem dla którego monolityczne ebuildy są jeszcze dostępne dla KDE 3.x.

Czy zamierzacie całkowicie usunąć stare (monolityczne) ebuildy?

Ostatecznie taki jest plan. Jednak, dla wszystkich wydań KDE 3.4 będą dostępne zarówno stare, jak i nowe wersje ebuildów.

Jeśli ktoś preferuje monolityczne ebuildy zamiast tych rozdzielonych, niech poda nam swoje powody.

Teraz jest tyle ebuildów! Jak mam odnaleźć ten, którego właśnie potrzebuję?

Więc, po pierwsze, jeśli wiesz, że pakiet jakiego szukasz był w kdebase, możesz ją otrzymać poprzez emerge kdebase-meta co da taki sam rezultat jak zemergowanie monolitycznego kdebase. A więc nie ma tu żadnych niedogodności w związku z nowymi ebuildami.

Oczywiście wszystkie standardowe metody wyszukiwania paczek także działają. To tak samo jak szukanie aplikacji Gnome... Wystarczy znajomość nazwy aplikacji, której się szuka.

Sytuacja mogłaby prawdopodobnie się poprawić dzięki wprowadzenie większej ilości -meta ebuildów. Są one tylko listami zależności, a więc nie kosztują wiele pracy. Jednak nie zdecydowaliśmy się jeszcze na to. Byłoby jednak miło gdyby Portage zyskało odpowiednią funkcjonalność zanim zajmiemy się tym szerzej.

Jak wylistować/odmiergować wszystkie rozdzielone ebuildy pochodzące z danej paczki?

Można to przetłumaczyć na wylistowanie wszystkich rozdzielonych ebuildów KDE pochodzących z, powiedzmy, monolitycznego ebuilda KDE. Jeszcze raz, odpowiednia implementacja (taka jak [GLEP 21](#)) sprawi, że będzie to trywialne. Jednak dzisiaj, musimy się zapoznać w pewnym stopniu z implementacją eclass KDE.

kde-functions.eclass definiuje funkcje zwane get-parent-package() oraz get-child-packages() które przeprowadzają tłumaczenie za użytkownika. Te dwie funkcje są poprawnym rozwiązaniem dla postawionego problemu i mogą zostać wykonane z jakiegoś ebuilda albo zewnętrznego skryptu bash. Na przykład:

Listing 3.1: Implementacja eclass KDE

```
$ function die() { echo $@; } # pokazuj błędy
$ source /usr/portage/eclass/kde-functions.eclass
$ get-parent-package konqueror # nie zadziała, potrzebna jest pełna nazwa
Package konqueror not found in KDE_DERIVATION_MAP, please report bug # zwrócenie błędu
$ get-parent-package kde-base/konqueror # pełnowartościowa nazwa pakietu
kde-base/kdebase # zwrócony wynik
$ get-child-packages kde-base/kdebase
# (długa lista paczek)
```

Jeśli skrypt nie został napisany w bashu należy przegrepować kde-functions.eclass w celu wyodrębnienia definicji zmiennej KDE_DERIVATION_MAP, której używają wyżej wymienione funkcje. Zmienna ta zawiera oddzieloną białymi znakami listę słów, każde dwa kolejne słowa przyporządkowują paczkę rodzica do rozdzielonego pliku ebuild - dziecka.

6. Instalacja i konfiguracja Gnome

1. Czym jest GNOME?

Projekt

Projekt GNOME to grupa osób zajmujących się rozwijaniem środowiska graficznego o tej samej nazwie. Fundacja GNOME koordynuje prace i zajmuje się niezwiązanymi z programowaniem aspektami rozwoju środowiska GNOME.

Oprogramowanie

GNOME to środowisko graficzne oraz platforma deweloperska i jest doskonały wybór zarówno dla użytkowników domowych jak i dla developerów.

Społeczność

Tak jak każdy duży projekt otwartego oprogramowania, GNOME posiada szeroką grupę użytkowników oraz developerów. Na stronie Footnotes znajdują się wiadomości dla zwykłych użytkowników. Osoby bardziej zaawansowane na pewno zainteresują się za to strona GnomePlanet, a zainteresowanych pomogą w rozwoju środowiska GNOME strona Developer.Gnome.Org.

2. Instalowanie GNOME

Co jest potrzebne?

Przed rozpoczęciem instalacji GNOME należy odpowiednio zmienić flagi USE w pliku /etc/make.conf tak, aby były włączone gtk* i gnome. Jeśli potrzebna jest obsługa hald, warstwy abstrakcji sprzętowej, należy dodać do listy flagę hal. Zmienna avahi doda GNOME obsługę wykrywania DNS (podobną do Rendevouz z Mac OS X). Jeśli nie jest potrzebna obsługa KDE (innego dużego środowiska graficznego), można usunąć z listy flagi qt, arts i kde.

Listing 2.1: Przykładowa lista flag USE w /etc/make.conf

```
USE="-qt -qt3 -qt4 -arts -kde gtk gtk2 gnome hal avahi"
```

Potem można zainstalować GNOME poprzez zemergowanie pakietów gnome i xscreensaver:

Listing 2.2: Instalowanie GNOME

```
# emerge gnome xscreensaver
```

Minimalne środowisko Gnome można znaleźć w pakiecie gnome-light:

Listing 2.3: Instalowanie minimalnego środowiska GNOME

```
# emerge gnome-light
```

Zajmie to chwilę. Po zakończeniu instalacji należy uaktualnić zmienne środowiskowe:

Listing 2.4: Uaktualnianie zmiennych środowiskowych

```
# env-update && source /etc/profile
```

Wynik polecenia emerge chwilę wcześniej zawierał sugestię dotyczącą dodania gamin na domyślny poziom uruchomieniowy w celu umożliwienia nautilusowi i gnome-vfs monitorowania zmian w plikach:

Listing 2.5: Instalowanie gamin, programu monitorującego zmiany w plikach

```
# emerge gamin
```

Ważne: Jeśli jest to zmiana z fam (programu, który był używany w tym samym celu w przeszłości, należy najpierw usunąć famd ze wszystkich poziomów uruchomieniowych, a następnie całkowicie odinstalować:

Listing 2.6: Zmiana na gamin

```
# rc-update del famd default
# emerge --unmerge fam
```

Następnie należy wyczyścić wszystkie pozostałe poziomy uruchomieniowe.

Listing 2.7: Dodawanie hald i avahi-dnsconfd na domyślny poziom uruchomieniowy

```
# /etc/init.d/hald start
# rc-update add hald default

# /etc/init.d/avahi-dnsconfd start
# rc-update add avahi-dnsconfd default
```

Pierwsze wrażenia

Po pierwsze warto dowiedzieć się co właściwie się zbudowało. Opuszczamy konto roota i logujemy się jako zwykły użytkownik. Skonfigurujemy system tak, aby uruchamiał GNOME po każdym wpisaniu polecenia startx. Więcej szczegółów na ten temat znajduje się w akapicie dotyczącym [używania startx](#) w dokumencie opisującym [konfigurację serwera X](#).

Listing 2.8: Konfigurowanie GNOME jako domyślnego środowiska

```
$ echo "exec gnome-session" > ~/.xinitrc
```

Następnie uruchamiamy środowisko za pomocą polecenia startx:

Listing 2.9: Uruchamianie GNOME

```
$ startx
```

Jeśli wszystko pójdzie dobrze uruchomi się środowisko graficzne z logo GNOME. Pora przyjrzeć się możliwym do przeprowadzenia na samym początku dodatkowym zmianom w konfiguracji.

3. Konfigurowanie GNOME

Graficzny menedżer logowania GNOME

Jeśli menedżer logowania GNOME (GDM) ma być uruchamiany automatycznie po starcie systemu tak, aby możliwe było zalogowanie się ze środowiska graficznego, należy dodać skrypt startowy xdm na domyślny poziom uruchomieniowy:

Listing 3.1: Dodawanie xdm na domyślny poziom uruchomieniowy

```
# rc-update add xdm default
```

Następnie należy otworzyć plik /etc/rc.conf i odpowiednio zmienić zawartość zmiennej DISPLAYMANAGER.

Listing 3.2: Edytowanie /etc/rc.conf

```
DISPLAYMANAGER="gdm"
```

Po ponownym uruchomieniu komputera pojawi się GDM i umożliwi zalogowanie się do środowiska GNOME. Istnieje opcja uruchamiania z tego menedżera innych środowisk, należy w tym celu wyedytować plik /usr/share/xsessions/. Warto wspomnieć, że jeśli korzysta się z GDM to niepotrzebny staje się plik ~/.xinitrc.

W celu uruchomienia hald należy uruchomić gnome-volume-manager i wyedytować ustawienia.

7. Instalacja i konfiguracja Fluxbox

1. Wstęp

Wprowadzenie do menedżera okien Fluxbox

Menedżery okien to programy, które są uruchamiane przez serwer X11 w celu zarządzania aplikacjami graficznymi. WM (skrót od ang. Window Manager) kontrolują gdzie jakie okno jest umieszczane, jego rozmiar, i które pulpity mają do niego dostęp. Takie zachowanie może być automatyczne, poprzez zbiór reguł zdefiniowanych przez użytkownika albo może być wykonywane w czasie pracy za pomocą myszki i/lub sktótów klawiszowych, w zależności od specyfiki WM.

Jednak niektóre WM dostarczają więcej funkcjonalności niż tylko rozmieszczenie okien. Fluxbox dostarcza także platformę do uruchamiania aplikacji, konfigurowania skrótów klawiszowych i obrazów tła jak również wyświetlanie informacji. Domyśły WM Gnome, Metacity, nie dostarcza tych funkcjonalności, ale są one dostępne przez inne programy w Gnome. Dlatego nie można powiedzieć, że Fluxbox jest bardzo minimalistycznym WM. Jednakże ponieważ Fluxbox zawiera wszystkie niezbędne komponenty środowiska graficznego na biurko, wiele osób używa go zamiast dużych środowisk jak KDE czy Gnome.

Żeby sprawę jeszcze bardziej skomplikować Fluxbox może zastąpić domyślny WM w Gnome czy w KDE, np. niektóre osoby uważają, że Metacity, menadżer okien w Gnome, po prostu nie ma wystarczającej elastyczności, której potrzebują, a Fluxbox nie ma wbudowanych i zintegrowanych wszystkich aplikacji, których wymagają od kompletnego środowiska graficznego na biurku. Taka osoba może używać Fluxboxa jako menadżera okien i Gnome dla całej reszty. Należy jednak zauważyć, że w takim przypadku wystąpią pewne problemy związane z nakładającymi się funkcjonalnościami.

Fluxbox może być także rozszerzony przez oprogramowanie zewnętrzne. Często jest spowodowane ograniczonymi zasobami sprzętowymi lub osobistymi preferencjami użytkownika.

Przewodnik ten skierowany jest do osób nie znających Fluxboxa, zainteresowanych nim oraz chcących go wykorzystać w swoim systemie. Pokażemy w nim jak za pomocą zewnętrznych aplikacji rozszerzyć Fluxboxa o wiele przydatnych funkcji.

2. Instalowanie Fluxboksa

Instalacja podstawowa

Dzięki odpowiedniemu ebuildowi instalacja Fluxboksa jest dość prosta i szybka. Poprzez dodanie flag USE gnome i kde można zapewnić zgodność naszego menedżera z jego konkurentami, z doświadczenia wiemy jednak, że nie jest to absolutnie do niczego potrzebne. Wszystkie aplikacje związane z KDE i GNOME można zemergować i skonfigurować osobno. Autor tego tekstu z powodzeniem używa np. k3b (programu z KDE) do nagrywania płyt pod fluxboksem.

Warto też zaopatrzyć się w program, który będzie blokował sesję środowiska X przed nieautoryzowanym dostępem - xlock. Większość użytkowników na pewno zechce również mieć możliwość zmiany tła pulpitu. W tym celu powinni emergować eterm, który jednocześnie jest ładnym terminalem dla X11.

Listing 2.1: Emergowanie Fluxbox

```
# emerge fluxbox xlockmore x11-terms/eterm
```

Cóż, to wszystko. Fluxbox jest już zainstalowany! Jednak z domyślnym zestawem narzędzi nie można w pełni korzystać z jego zalet. Kolejne rozdziałły pomogą w zainstalowaniu innych, potencjalnie potrzebnych pakietów oraz w skonfigurowaniu Fluxboksa i zewnętrznych programów, aby dobrze pracowały dla użytkownika. Należy pamiętać o tym, że żaden z opisanych kroków nie jest obowiązkowy, dlatego też można w nich dowolnie przebierać.

3. Konfigurowanie Fluxboksa

Przygotowanie środowiska X11

Fluxbox można traktować jak każdą inną aplikację uruchamianą w środowisku X11 i uruchamiać poprzez wpisanie w konsoli startx, a następnie fluxbox w terminalu serwera X. Jest to jednak dość niewygodna metoda. Możliwe jest ustawienie automatycznego startowania fluxboksa przy każdym uruchomieniu serwera X. Dokonuje się tego poprzez wydanie (z konta zwykłego użytkownika) polecenia:

Listing 3.1: Edytowanie pliku xinit

```
$ echo exec startfluxbox > ~/.xinitrc
```

Takie rozwiązanie nie jest jednak wygodne dla większości użytkowników, którzy przez cały czas chcą pracować w środowisku graficznym. Wpiswanie startx za każdym razem po starcie komputera, aby wydostać się ze środowiska graficznego jest bardzo uciążliwe. Takie postępowanie stanowi też potencjalne zagrożenie dla nieświadomych użytkowników. Jeżeli ktoś zablokował ekran za pomocą xlock, ale środowisko X11 uruchomiło z konsoli, to ktoś inny, kto posiada fizyczny dostęp do komputera może przełączyć się na konsolę z której uruchomiono sesję X11, zabić ją i może używać konta użytkownika. Aby zapobiec takiej sytuacji można albo uruchomić sesję X11 w tle, albo uruchomić ją w sesji programu screen, po czym odłączyć screen i wylogować się. Delikatnie mówiąc jest to zadanie dosyć żmudne. Problem ten nie istnieje jeżeli używany jest graficzny menedżer logowania. W takim jednak przypadku

sesja X11 będzie uruchomiona z konta superużytkownika (root) co również może nie być dobrym rozwiązaniem. W momencie powstawania tego przewodnika autor nie posiada wiedzy na temat istnienia jakiegokolwiek exploitu. Jeżeli jednak ktoś martwi się tym zagadnieniem to może albo nie używać wcale środowiska graficznego, albo stosować podane wyżej rozwiązanie ze screenem. Należy przy tym mieć na uwadze fakt, że nie jest to kwestia Fluxboksa, lecz samego środowiska X11.

Mamy do wyboru kilka graficznych menedżerów logowania do wyboru. W przewodniku omówimy gdm. Można też korzystać z innych menedżerów logowania, ale to gdm najlepiej pracuje zarówno z Fluxboksem jak i z innymi menedżerami okien zainstalowanymi w komputerze użytkownika takimi jak np. GNOME, KDE czy xfce.

Listing 3.2: Menedżer logowania

```
# emerge gdm
# rc-update add xdm default
```

Ważne: Należy zwrócić uwagę na to, by najpierw zainstalować gdm, a dopiero potem dodawać skrypt xdm do poziomu startowego. Odwrócenie kolejności wykonania tych poleceń zakończy się niepowodzeniem. Poza tym należy zmodyfikować jeden systemowy plik konfiguracyjny.

Listing 3.3: Edycja /etc/rc.conf

```
DISPLAYMANAGER="gdm"
```

Wystroje pulpitu i grafika

Wszyscy, którzy chcą mieć minimalną wersję Fluxboksa mogą ten rozdział pominąć. Wszystkim innym wypada jednakże umożliwić zmianę wyglądu menedżera okien. W tym celu zainstalowane zostaną schematy wyglądu specyficzne dla Fluxboksa, grafiki związane z Gentoo do wykorzystania w dowolnym menedżerze okien, oraz schematy wyglądu do wykorzystania w każdym menedżerze okien serii *box.

Listing 3.4: Instalowanie wystrojów i grafik

```
# emerge commonbox-styles commonbox-styles-extra \
  fluxbox-styles-fluxmod gentoo-artwork
```

Kolorowanie składni Fluxboksa w Vimie

Następnie zainstalowany zostanie specjalny schemat kolorowania składni dla edytora vim. Umożliwia on łatwiejsze i bardziej czytelne edytowanie zawartości specjalnych plików konfiguracyjnych Fluxboksa takich jak pliki keys i init. Jest to bardzo mała wtyczka dla vima dlatego też zaleca się jej zainstalowanie (no chyba, że jesteśmy jednym z fanów emacs, wtedy zalecane jest emerge -C emacs && emerge vim gvim) ;-).

Listing 3.5: Emergowanie kolorowania składni Fluxboksa

```
# emerge fluxbox-syntax
```

Uruchamianie aplikacji

Każdy menedżer okien wyposażony jest (albo powinien być wyposażony) w specjalny program, za pomocą którego można uruchamiać programy bez konieczności otwierania konsoli specjalnie do tego celu. We Fluxboksie takim specjalnym programem jest fbrun i jest on zawsze zainstalowany wraz z Fluxboksem. Program ten można wywołać poprzez wpisanie w linii poleceń fbrun, jednak uruchamianie z konsoli programu fbrun po to, aby za jego pomocą uruchomić inny program jest jeszcze gorszym rozwiązaniem niż po prostu uruchomić ten inny program wprost z konsoli. Dlatego też powinno się zdefiniować klawisz skrótu albo dodać odpowiednią pozycję do menu, która będzie uruchamiała fbrun, albo wykonać obydwie te rzeczy.

Aby dodać do menu pozycję "Uruchom..." należy do pliku `~/.fluxbox/menu` dopisać linię:

Listing 3.6: Dodawanie pozycji w menu uruchamiającej fbrun

```
[exec] (Uruchom...) {fbrun}
```

Więcej informacji na temat menu Fluxboksa znajduje się w dalszej części tego przewodnika.

Więcej informacji, takich jak na przykład opcje uruchamiania, można uzyskać czytając instrukcję lub wpisując man fbrun. Należy przy tym pamiętać, że w celu wykorzystania opcji uruchamiania trzeba będzie ponownie edytować plik keys lub menu, aby te opcje wprowadzić.

Klawisze skrótów we Fluxboksie

Standardowo Fluxbox wyposażony jest w bardzo ubogi zestaw klawiszy skrótu. Pozwalają one jedynie na przełączanie między oknami oraz między pulpitami. Zanim jednak zaczniemy modyfikować ten zestaw, należy wiedzieć kilka rzeczy. W pliku keys jest kilka dziwnych modyfikatorów. Mod1 to powszechnie znany klawisz "Alt", a Mod4 jest klawiszem bez nazwy (klawisz windows). Control i Shift to po prostu klawisze Control i Shift. Lepiej jest jednak stosować klawisze Alt/Windows ponieważ inne aplikacje środowiska graficznego zwyczajowo korzystają z klawiszy Control i Shift dla swoich własnych klawiszy skrótu.

Przedstawione przykłady są zupełnie dowolne. Użytkownik może dowolnie modyfikować je w zależności od swoich potrzeb. Przewodnik ten uwzględnia programy, które według autora są najczęściej używane, dlatego też również nazwy programów można zastąpić tymi, które są stosowane przez użytkownika. Więcej informacji można znaleźć w instrukcji Fluxboksa.

Mając tą świadomość, należy uruchomić edytor tekstu (jako zwykły użytkownik, nie root), otworzyć plik `~/.fluxbox/keys` i zacząć udoskonalać klawisze skrótu!

Listing 3.7: Modyfikowanie skrótów klawiaturowych.

```
#blokuje sesję X11 dla bieżącego użytkownika
Mod4 l :ExecCommand xlock

# uruchamianie różnych programów. To są tylko przykłady dla zobrazowania zastosowania.

Mod1 f :ExecCommand firefox
Mod1 t :ExecCommand thunderbird
Mod1 o :ExecCommand oowriter
Mod1 v :ExecCommand gvim

# Fluxbox nie posiada funkcji regulacji głośności dźwięku. Należy więc
# ją dodać.
# (przyjęto, że używana jest ALSA. Jeżeli wykorzystywany jest oss, to jest się
# zdanym na samego siebie)
Mod1 e :ExecCommand Eterm -name alsa -e alsamixer

# Uruchomienie fbrun, który jednorazowo uruchamia inne programy.
Mod1 r :ExecCommand fbrun

# Programy pod kontrolą. To po prostu ułatwia życie...
Mod1 4 :Close
Mod1 m :Minimize

# jeżeli zainstalowany jest aterm (emerge x11-terms/aterm), to to
# polecenie otworzy ładny terminal, przezroczysty, bez ramki, dobrze
# wpasowujący się w ekran o rozdzielcości 1024 x 768, pozostawiając miejsce na
# conky. Polecenie można dowolnie modyfikować.
Mod1 a :ExecCommand aterm -name aterm -sl 3000 -tr +sb -sr -sk -bg black -fg \
white -fade 90 -bl -tn xterm -fn \
-misc-fixed-medium-r-normal-*-*-120-*-*-c-*--iso8859-2 -g 116x57

# tak jak powyżej, ale otwierany jest terminal dla roota. Należy
# oczywiście znać hasło roota, aby z niego skorzystać.
Mod1 s :ExecCommand aterm -name aterm -sl 3000 -tr +sb -sr -sk -bg black -fg \
white -fade 90 -bl -tn xterm -fn \
-misc-fixed-medium-r-normal-*-*-120-*-*-c-*--iso8859-2 -g 116x57 -e su -
```

Monitoring systemu

Powszechnym życzeniem użytkowników jest posiadanie monitora systemu w środowisku graficznym. Najbardziej popularnym narzędziem służącym do monitoringu systemu jest Gkrellm, ale zgodnie z zasadą minimalizmu jakiej hołduje Fluxbox, zastosowany zostanie prostszy program: conky. Warto pamiętać, że w Internecie znajduje się znacznie więcej programów tego typu.

Listing 3.8: Instalowanie conky

```
# emerge conky
```

Domyślny plik konfiguracyjny dla Torsmo jest kiepsko opisany, ale więcej informacji można znaleźć w instrukcji do programu lub czytając tekst Konfiguracja conky.

Ikony

Standardowo Fluxbox nie jest wyposażony w żadne narzędzie do wyświetlania ikon na pulpicie. Wszystkie czynności załatwiane są za pomocą klawiszy skrótu, lub za pomocą menu wywoływanego kliknięciem prawym klawiszem myszy. iDesk to program pozwalający na zaspokojenie tej potrzeby, przeznaczony dla użytkowników z minimalistycznymi menedżerami okien i umożliwiający wyświetlanie ikon w głównym oknie. Posiada też kilka świetnych dodatkowych funkcji, na przykład związanych z przezroczystością. Pomijając nazwę, produkt ten nie ma nic wspólnego z Apple i Macintoshami (a przynajmniej nic o tym nie wie autor artykułu). Instalowanie iDesk jest proste:

Listing 3.9: Instalowanie iDesk

```
# emerge idesk
```

Po zainstalowaniu należy wybrać zestaw ikon do wyświetlania. iDesk informacje o wyświetlanych ikonach przechowuje w pliku `~/.ideskrc`. Dodawanie ikon jest zadaniem raczej pracochłonnym i nie może zostać dobrze opisane w tym przewodniku. Aby uzyskać więcej informacji można przeczytać man `idesk` lub skorzystać z następujących stron: iDesk's Usage Guide oraz tego wątku na oficjalnym forum Gentoo.

Graficzny menedżer plików.

Należy zauważyć, że Gentoo nie potrzebuje graficznego menedżera plików. Wedle skromnej opinii autora, możliwym i w istocie prostszym jest przeprowadzanie wszelkich operacji na plikach z linii poleceń. Nie każdy jednak musi się z tym zgadzać...

Wielu nowych użytkowników Gentoo/Fluxboksa jest zniechęconych brakiem graficznego menedżera plików takiego jak np. nautilus w Gnome. Fluxbox sam w sobie nie posiada takiego menedżera, ale jak zwykle, bardzo łatwo można doinstalarować taki menedżer plików. Wybór jest bardzo szeroki, ale w tym przewodniku ograniczymy się do dwóch. Jeden dla minimalistów, którzy lubią proste tekstowe podejście i drugi dla wszystkich, którzy lubią kliknąć w ikony. Można oczywiście zainstalować i wypróbować obydwa.

Menedżer plików gentoo - minimalizm

Gentoo jest menedżerem plików, który posiada wszelkie atuty graficznego menedżera plików, ale pozbawiony jest tych wszystkich kolorowych wodotrysków jakie zwyczajowo są dołączane do takiego programu.

Listing 3.10: Instalowanie graficznego menedżera plików

```
# emerge gentoo
```

Uwaga: Menedżer plików Gentoo jest projektem zupełnie niezwiązanym z dystrybucją Gentoo Linux. Więcej informacji na ten temat można znaleźć w Name and Logo Usage Guidelines.

Z założenia menedżer plików Gentoo ma być w pełni konfigurowalny, a więc można go dowolnie dopasować do własnych potrzeb.

Menedżer plików Rox - zawiera ikony

Menedżer plików Rox korzysta z ikon oraz tekstu i w bardzo dużym stopniu przypomina Eksploratora Windows.

Listing 3.11: Instalowanie menedżera plików Rox

```
# emerge rox
```

Działanie menedżera plików Rox jest w zasadzie takie jak wielu innych tradycyjnych graficznych menedżerów plików, a więc jego obsługa powinna być intuicyjna. Jeżeli tak nie będzie to zawsze można skorzystać z instrukcji do programu.

Przeglądarka obrazów

Wiele menedżerów okien wyposażonych jest w narzędzie do przeglądania obrazów. Są to małe, nie rozbudowane aplikacje, które nie dają możliwości edytowania plików (co z kolei jest zadaniem aplikacji takich jak gimp i podobnych). Aplikacja taka jest konieczna

jeżeli korzystamy tylko z linii poleceń lub z menedżera plików takiego jak gentoo lub rox.

Listing 3.12: Instalowanie gqview

```
# emerge gqview
```

gqview może zostać uruchomiony w każdym katalogu i można nim przeglądać obrazy we wszystkich obsługiwanych formatach.

Ustawienie schematu wyglądu, tła i skryptu startowego

Fluxbox posiada wiele schematów wyglądu. Są one dostępne z menu wywoływanego prawym kliknięciem myszy, w pozycji "Fluxbox menu", a potem "System Styles". Każdy z nich ustawia własne tło, które jest albo jednolitym kolorem, albo jakimś rodzajem gradientu koloru. Bez względu na to jaki schemat wyglądu jest wybrany to i tak większość użytkowników woli ustawić własny typ tła. Aby to zrobić należy podać edycji kolejny plik konfiguracyjny Fluxboksa, który odpowiada za to co dzieje się podczas startu Fluxboksa. Należy więc uruchomić edytor tekstu i otworzyć plik `~/.fluxbox/startup`.

Jeżeli plik ten już istnieje to należy wykasować jego zawartość. Do nowo utworzonego pliku należy dodać poniżej przedstawioną treść, usuwając wedle potrzeb znaki komentarza z interesujących użytkownika linii oraz uzupełniając wartości dla pól wypełnionych DUŻYMI LITERAMI.

Listing 3.13: Edytowanie skryptu startowego Fluxboksa.

```
# Gentoo's Fluxbox startup script

# Programs which need to run constantly, as opposed to a one time execution,
# need "&" at the end of the command.

# Show the Fluxbox splash-screen
#fbsetbg -C /usr/share/fluxbox/splash.jpg

# This sets a background. You must have a program setup to do this
# (x11-terms/eterm is recommended)
#fbsetbg -f PATH_TO_IMAGE

# Custom fonts directory
#xset +fp PATH_TO_FONT_DIRECTORY

# Starts the icons program
#idesk &

# This MUST be the last command!
exec /usr/bin/fluxbox -log ~/.fluxbox/log
```

Generowanie menu.

Do stworzenia menu autor używa edytora tekstu. Jeżeli takie podejście jest interesujące dla użytkownika to można pominąć ten rozdział i przejść od razu do rozdziału Ręczne edytowanie menu..

Po utworzeniu menu za pomocą automatycznego generatora (patrz następny rozdział) można edytować `~/.fluxbox/menu` ręcznie, aby dodać pozycje, lub usunąć programy, z których się nie korzysta. Zauważyc przy tym należy, że usunięcie z menu określonej pozycji nie usuwa z systemu odpowiadającej tej pozycji aplikacji. Dana aplikacja nadal może być uruchomiona z linii poleceń.

W systemie, w którym zainstalowano Fluxboksa jest też program o nazwie `fluxbox-generate_menu`, który służy do automatycznego stworzenia menu. Aby go używać należy posiadać Fluxbox 0.9.13 lub nowszy. Pewność, że jest to odpowiednia wersja uzyskamy po wydaniu polecenia `emerge --sync && emerge -u fluxbox`. Zainstaluje się dzięki temu specjalna wersja skryptu, zawierająca poprawki dla Gentoo, które umożliwiają mu odnalezienie większej ilości programów niż wcześniej. Skrypt dzięki temu staje się znacznie praktyczniejszy niż poprzednia jego wersja, pozwala na odnalezienie w systemie programów, o których istnieniu użytkownik nie ma pojęcia. Uruchamiamy go w poniższy sposób (z konta zwykłego użytkownika):

Listing 3.14: Generowanie menu dla wersji >=fluxbox-0.9.13

```
$ fluxbox-generate_menu -is -ds
```

Polecamy uaktualnienie i skorzystanie z polecenia fluxbox-generate_menu -is -ds zamiast pracy z innymi programami tego typu, jak np. mmaker, ponieważ nie są one dostosowane do struktury katalogów w Gentoo.

Uwaga: Po zastosowaniu tej metody, która automatycznie generuje menu, nadal istnieje możliwość ręcznego sterowania zawartością menu. Ręczne utworzenie i edycja `~/.fluxbox/usermenu` (patrz następny rozdział) pozwala na stworzenie menu dopasowanego do wymagań użytkownika, które nie zostanie skasowane przy następnym uruchomieniu skryptu `fluxbox-generate_menu`.

Ręczne edytowanie menu.

Menu Fluxboksa znajduje się w pliku tekstowym o bardzo prostej składni opisanej w tym rozdziale. Domyślnym miejscem przechowywania pliku menu jest `~/.fluxbox/menu`.

Pozycje w menu, które uruchamiają aplikacje definiowane są za pomocą elementu `[exec]`. Pozycje menu definiuje się po elemencie `[begin]` (pozycje główne), lub między elementami `[submenu]` i `[end]` (podmenu), np.:

Listing 3.15: Definiowanie nowej pozycji w menu.

```
# Custom fluxbox menu
[begin] (Fluxbox)
(...)
[exec] (XTerm White on Black) {xterm -ls -fg white -bg black}
(...
[submenu] (More terminals)
    [exec] (Aterm default) {aterm}
    [exec] (Rxvt default) {rxvt}
[end]
(...
[end]
```

Główne pozycje menu zaczynają się tagiem `[begin]`, po którym w nawiasach umieszczona jest nazwa menu. Główne menu kończy się tagiem `[end]`. Wszystkie polecenia menu (zwane tutaj tagami) zamknięte są w nawiasy prostokątne, nazwy pozycji w nawiasy, a właściwe polecenie zamknięte jest w nawiasy klamrowe. Składnia pliku dopuszcza stosowanie komentarzy w postaci linii rozpoczętych znakiem `#`.

W powyższym przykładzie pierwsza linia z elementem `[exec]` definiuje nową pozycję w menu o nazwie "XTerm White on Black", która będzie uruchamiała polecenie xterm -ls -fg white -bg black zupełnie tak samo jak gdyby to miało miejsce w linii polecień. Dalej mamy podmenu zdefiniowane za pomocą tagu `[submenu]` o nazwie "More terminals", w którym znajdują się dwie kolejne pozycje z poleceniami `[exec]`. Każda definicja podmenu musi być zakończona tagiem `[end]`.

Istnieją też inne polecenia, które można zastosować w pliku `~/.fluxbox/menu`. Więcej informacji na ten temat można znaleźć w [oficjalnej dokumentacji](#).

8. Przewodnik po sprzętowej akceleracji 3D

1. Wprowadzenie

Sprzętowa akceleracja 3D - czym jest i po co?

Dzięki sprzętowemu przyspieszeniu 3D proces trójwymiarowego renderowania wykorzystuje procesor graficzny na karcie graficznej do rysowania trójwymiarowych obrazów zamiast zużywania cennych zasobów procesora głównego. Nazywa się to "sprzętową akceleracją", w odróżnieniu od "akceleracji programowej", ponieważ bez przyspieszenia 3D procesor główny musiałby rysować wszystko samodzielnie używając bibliotek renderowania Mesa, co pochłania dużo mocy obliczeniowej. Podeczas gdy Xorg typowo obsługuje dwuwymiarową akcelerację sprzętową, często nie ma sprzętowej akceleracji 3D. Trójwymiarowe sprzętowe przyspieszenie jest cenne w sytuacjach wymagających przedstawiania trójwymiarowych obiektów, np. w grach lub trójwymiarowym modelowaniu, czy też w 3D CAD.

Jak uruchomić sprzętową akcelerację 3D?

W wielu przypadkach istnieją zarówno sterowniki binarne, jak i o otwartym źródle. Wypada preferować te drugie, skoro używamy Linuksa i otwartość oprogramowania stanowi jedną z podstawowych zasad. Czasami jednak nie mamy wyboru i musimy użyć sterowników binarnych, jak w przypadku kart nVidii. Sterowniki binarne to na przykład x11-drivers/nvidia-drivers i x11-drivers/nvidia-legacy-drivers dla kart nVidia oraz media-video/ati-drivers dla kart ATI. Inne sterowniki open-source to media-video/kyro-kernel dla kart KyroII i media-video/ati-gatos dla kart ATI, które mają na celu lepiej wspierać możliwości wideo ATI.

Czym jest DRI?

Direct Rendering Infrastructure), czyli Infrastruktura Bezpośredniego Odtwarzania, znana jako DRI, jest szkieletem dla bezpośredniego dostępu do sprzętu graficznego w sposób wydajny i bezpieczny. Zawiera modyfikacje dla serwera X, bibliotek klient-serwer oraz jądra. Najważniejszym zastosowaniem DRI jest tworzenie środowiska do szybkiej implementacji OpenGL.

Co to jest X11-DRM i jaki ma związek z Xorg?

X11-DRM jest rozszerzeniem Xorga, które dodaje akcelerację 3D dla kart graficznych poprzez dołączenie do jądra systemu modułu niezbędnego do bezpośredniego renderingu.

Do czego służy ten przewodnik

Przewodnik ten jest dla tych, którzy nie mogą uzyskać bezpośredniego renderingu z samym serwerem Xorg. X11-DRM działa ze sterownikami serii 3dfx, gamma, i8x0, matrox, rage128, radeon, mach64 oraz sis300. Właściciele kart VIA powinni zamiast x11-drm używać DRI znajdującego się w nowszych wersjach jądra, (np. >2.6.13). Więcej informacji można znaleźć na [stronie DRI](#).

Pytania, sugestie, propozycje

Wszelkie pytania, sugestie i propozycje prosimy zgłaszać na adres Donniego Berkholza.

2. Instalowanie Xorg i konfiguracja jądra systemu

Instalowanie Xorg

Aby uruchomić Xorg, należy przeczytać Przewodnik po konfiguracji Xorg.

Konfiguracja jądra

Należy zaznaczyć odpowiednią dla posiadanego chipsetu opcję.

Listing 2.1: Sprawdzanie chipsetu AGP

```
# emerge pciutils; lspci | grep AGP
# 00:01.0 PCI bridge: Intel Corp. 440BX/ZX/DX - 82443BX/ZX/DX AGP bridge (rev 03)
(Wyniki sprawdzenia mogą być różne zależnie od sprzętu.)
```

Jeżeli jądra nie jest w stanie obsługiwać chipsetu ciągle jest szansa uruchomienia sprzętu poprzez użycie parametru `agp=try_unsupported`. Spowoduje on użycie rdzennych algorytmów Intel'a do obsługi AGP. Aby dodać ten parametr do konfiguracji jądra, należy wyedytować plik konfiguracyjny bootloadera.

Poniższe opcje dotyczą większości, jeżeli nie wszystkich typów jądra systemu. Zostały one skonfigurowane z użyciem gentoo-sources-2.4.20-r5.

Listing 2.2: Konfiguracja jądra systemu.

```
# ls -l /usr/src/linux
lrwxrwxrwx 1 root root 22 May 29 18:20 /usr/src/linux -> linux-2.4.20-gentoo-r5
(Należy się upewnić, że /usr/src/linux to link symboliczny do aktualnego kernela.)
# cd /usr/src/linux
# make menuconfig
```

Listing 2.3: Opcje w make menuconfig

```

Processor type and features --->
<*> MTRR (Memory Type Range Register) support
Character devices --->
<M> /dev/agpgart (AGP Support)
[M] Intel 440LX/BX/GX and I815/I820/I830M/I830MP/I840/I845/I850/I860 support
(Należy wybrać swój chipset zamiast powyższego)
[ ] Direct Rendering Manager (XFree86 DRI support)

```

Posiadacze jądra serii 2.4 muszą się upewnić, że Direct Rendering Manager (DRM) jest wyłączony (off). Pakiet X11-DRM sam to ustawia. Użytkownicy kernela serii 2.6 powinni uruchomić (enable) DRM ręcznie, ponieważ jak dotychczas pakiet X11-DRM nie jest obsługiwany w serii 2.6.

Kompilowanie i instalacja jądra systemu

Listing 2.4: Kompilacja i instalacja jądra systemu

```

(Przykład odnosi się do jądra z serii 2.4)
# make dep && make clean bzImage modules modules_install
# mount /boot
# cp arch/i386/boot/bzImage /boot

```

Aby zmienić nazwę obrazu na inną niż "bzImage" należy po prostu skopiować obraz do "/boot/INNA_NAZWA" zamiast do "/boot". Trzeba również pamiętać o poprawieniu pliku konfiguracyjnego bootloadera (lilo.conf lub grub.conf). W przypadku LILO, trzeba jeszcze wczytać konfigurację przy pomocy polecenia /sbin/lilo.

3. Instalowanie X11-DRM i konfiguracja direct rendering

Instalowanie X11-DRM

Listing 3.1: Instalowanie X11-DRM

```
# emerge x11-drm
```

Konfigurowanie xorg.conf

Na niektórych chipsetach konieczne będzie przebudowanie pakietu xorg-x11 z flagą USE="insecure-drivers". Dotyczy to układów mach64, unichrome i savage na xorg-x11-6.8.2 oraz mach64 i unichrome na xorg-x11-6.8.99.x. Posiadacze układów Savage nie powinni korzystać z xorg-x11-6.8.99.x, ponieważ jego obsługa tam jest zepsuta.

Listing 3.2: Ponowna komplikacja xorg-x11

```

(Należy dodać flagę USE insecure-drivers, jeżeli używamy układów
wspomnianych wyżej)
# echo "x11-base/xorg-x11 insecure-drivers" >> /etc/portage/package.use
# emerge xorg-x11

```

Następny krok to otwarcie /etc/X11/xorg.conf w edytorze tekstu i dokonanie w nim zmian aktywujących DRI oraz GLX.

Listing 3.3: xorg.conf

```

...
Section "Module"
    Load "dri"
    Load "glx"
    ...
EndSection

```

```
...
Section "Device"
    Driver "radeon"
    ...
EndSection
...
Section "dri"
    Mode 0666
EndSection
```

Użytkownicy innych sterowników muszą odpowiednio zmodyfikować wpis "radeon".

Ladowanie modułów poprzez autoload.d

Moduł będzie automatycznie ładowany przy każdym starcie systemu jeśli dodamy jego nazwę do pliku /etc/modules.autoload.d/kernel-2.6.

Listing 3.4: Edytowanie /etc/modules.autoload.d/kernel-2.6

(Należy wpisać nazwę modułu, który ma być automatycznie wczytywany)
intel-agp

Uwaga: Jeżeli agpgart zostało skompilowane jako moduł należy je również dodać do /etc/modules.autoload.d/kernel-2.6.

4. Testowanie akceleracji 3D

Ponowne uruchomienie systemu z użyciem nowego jądra

Po ponownym uruchomieniu systemu i zalogowaniu się na „normalnego” użytkownika należy sprawdzić czy direct rendering został zainstalowany i czy dobrze działa.

Listing 4.1: Testowanie renderingu

```
$ startx
(Nie ma potrzeby ręcznego ładowania modułów dla agpart, powinny wczytać się automatycznie
$ glxinfo | grep rendering
direct rendering: Yes
(Jeżeli odpowiedź brzmi: "No", akceleracja 3D nie została zainstalowana)
$ glxgears
(Test ilości ramek na sekundę (FPS) w domyślnej wielkości. Liczba powinna być)
(znaczaco wyzsza niж przed instalacj± x11-drm. Test nalezy wykonać w czasie, kiedy
procesor mozliwie nie jest zajety innymi zadaniami.)
```

5. Używanie źródeł z CVS

Ostrzeżenie: Nie należy ich używać jeżeli pakiet działa prawidłowo.

Czy CVS jest potrzebny?

Po pierwsze należy sprawdzić czy pakiet x11-drm działa. Jeżeli nie działa, a przegląd logów wykazał, że nie jest to błąd konfiguracji, może się okazać konieczne wykorzystanie źródeł CVS. Można tam również pobrać aktualną na dany dzień wersję sterownika.

Czy źródła CVS obsługują daną kartę?

Należy sprawdzić na liście kart obsługiwanych przez DRI, czy dana karta się na niej znajduje. Jeżeli nie ma akurat naszej, może wystarczyć podobna.

Podążanie za instrukcjami CVS

Projekt DRI posiada dokumentację na temat komplikowania CVS. Należy wybrać odpowiednią instrukcję: Instalacja dla XFree86 lub Instalacja dla X.org.

Instalowanie CVS

Należy sprawdzić czy moduły DRI kernela zostały zbudowane:

Listing 5.1: Weryfikacja

```
# cd ~/DRI-CVS/build/xc/programs/Xserver/hw/xfree86/os-support/linux/drm/kernel; ls
```

Dla 3dfx Voodoo powinno być tdfx.o. W przypadku Matroxa G200/G400, mga.o. Dla Ati Rage 128, powinno być r128.o. W przypadku Ati Radeona radeon.o. Dla Intela i810, powinno być i810.o. Jeżeli nie udało się zbudować modułu DRI kernela należy sprawdzić, czy używana jest właściwa wersja jądra Linux. Zwykle najnowsze jądra nie obsługują DRI.

Kolejny krok to instalacja na XFree86 lub X.org. Po pierwsze trzeba wykonać kopię zapasową xfree lub xorg-x11.

Listing 5.2: Kopia zapasowa Xorg

```
# quickpkg xorg-x11
(Tak się wykonuje kopię zapasową xorg-11)
# make install
```

Powrót do instrukcji "Konfiguracja Xorg" powyżej.

Aby załadować właściwy moduł DRM do działającego jądra systemu, należy skopiować moduł kernela do /lib/modules/`uname -r`/kernel/drivers/char/drm/. Następnie uruchomić modules-update i zrestartować serwer X. Jeżeli nie jest uruchamiany kernel z wbudowanym tym elementem, zamiast `uname -r` należy użyć nazwy jądra.

Ostrzeżenie: Należy się upewnić że wszystkie wcześniejsze wersje modułów DRI zostały odłączone. Z drugiej strony niektóre moduły DRM wymagają wcześniejszego załadowania agpart.

6. Ulepszanie działania

Przyspieszanie direct renderingu

Kilka odpowiednio ustawionych opcji może zwiększyć wydajność nawet o 30 procent. Ustawia się je w /etc/X11/xorg.conf. Jednakże najpierw musimy się dowiedzieć czy wspierają je nasza płyta główna oraz karta graficzna.

Po pierwsze, sprawdzimy czy nasza karta graficzna wspiera fast write. Dokonamy tego poprzez dogłębną analizę wyniku polecenia lspci, a zwłaszcza informacji "VGA compatible controller".

Listing 6.1: Sprawdzanie karty graficznej

```
# lspci -vv
01:00.0 VGA compatible controller: ATI Technologies Inc Radeon Mobility M6
LY (prog-if 00 [VGA])
...
Capabilities: [58] AGP version 2.0
    Status: RQ=48 Iso- ArqSz=0 Cal=0 SBA+ ITACoh- GART64- HTrans- 64bit- FW+ AGP3- Rate=x1, x2, x4, x8, x16
```

Mimo, iż wypisywanych jest dużo informacji, obchodzi nas tylko **FW** w podsekcji "Status" dla AGP "Capabilities". Jeżeli otrzymujemy **FW+** oznacza to, że nasza karta wspiera fast write. Możemy przystąpić zatem do sprawdzania, czy poradzi z nim sobie nasza płyta główna.

Ważne: Jeżeli zamiast FW+ otrzymujemy FW- nie możemy odblokować fast write w xorg.conf, ponieważ nasza karta go nie obsługuje.

Teraz możemy sprawdzić, czy płyta główna obsługuje fast write. Znów posłużymy się poleceniem lspci, lecz tym razem należy zwrócić uwagę na sekcję "Host bridge"

Listing 6.2: Sprawdzanie płyty głównej

```
# lspci -vv
00:00.0 Host bridge: Intel Corporation 82830 830 Chipset Host Bridge
(rev 02)
...
Capabilities: [a0] AGP version 2.0
Status: RQ=32 Iso- ArqSz=0 Cal=0 SBA+ ITACoh- GART64- HTrans- 64bit- FW+ AGP3- Rate=x1,x2,x4
```

I tym razem należy przyjrzeć się opcji **FW**. Jeżeli zobaczymy **FW+** oznacza to, iż nasza płyta główna wspiera technologię fast write.

Ważne: Należy pamiętać, że zarówno informacja o karcie graficznej, jak i płycie głównej, musi pokazywać dokładnie ten sam stan FW. Jeżeli którykolwiek z urządzeń pokazuje FW- nie możemy uaktywnić fast write w naszym xorg.conf

Reasumując, jeżeli nasza płyta główna oraz karta graficzna wspierają fast write, możemy ustawić tę opcję w /etc/X11/xorg.conf i wyciągnąć to co najlepsze z naszego sprzętu :).

Listing 6.3: xorg.conf

```
Section "Device"
    Option      "AGPMode" "4"
    (Tak zwiększa się ilość FPS z 609 do 618.)
    Option      "AGPFastWrite" "True"
    (Ta opcja nie daje wymiernego efektu, za to może zdestabilizować system)
    (Może się okazać konieczne ustawienie jej również w BIOS-ie)
    Option      "EnablePageFlip" "True"
    (Ta opcja zwiększa ilość FPS z 618 do 702. Jej użycie jest również ryzykowne)
    ...
EndSection
```

Ostrzeżenie: Wybór opcji AGPFastWrite na chipsetach VIA może spowodować niespodziewane zawieszanie się systemu. Dzieje się tak, ponieważ chipsety VIA nie współpracują dobrze z fast write, zatem ustawiamy tę opcję na własne ryzyko.

Uwaga: Należy pamiętać, że jeżeli chcemy aby fast write działał poprawnie musimy odblokować odpowiednią opcję w BIOSie.

Aby poznać więcej opcji należy udać się pod adres tabeli możliwości na stronie DRI lub przeczytać ich listę na stronie Sourceforge.

7. Rozwiązywanie problemów

Sterowniki przestały działać po wymianie jądra na nowsze

Za każdym razem gdy rekompiluje się jądro lub nawet przełącza się na nowe, należy przebudować również jego moduły. Nie ma potrzeby ponownego instalowania xfree, czy xorg-11, ale niezbędne będzie przemergowanie x11-drm.

Nie działa rendering, nie wiem dlaczego

Spróbuj insmod radeon przed wystartowaniem serwera X. Spróbuj również wbudować agpgart do kernela, zamiast ładować go jako moduł.

Po wpisaniu startx, wyskakuje błąd: "[drm] failed to load kernel module agpgart"

Spowodowane jest to tym, że agpart został wkompilowany do jądra, zamiast być użytym jako moduł. Jeśli nie występują inne problemy można spokojnie zignorować ten komunikat.

Direct rendering nie działa, a w /var/log/Xorg.0.log widzę komunikat błędu o zbyt starej wersji sterownika

Nie używasz sterownika x11-drm. Sprawdź, czy masz skompilowany DRM oraz sterownik w jądrze. Prawdopodobnie nie masz.

Mam Radeona, chcę mieć TV-Out

Spróbuj sterowników ati-gatos. emerge -s gatos.

To nie działa. Moja karta jest bardzo nowa i w ogóle nie jest wspierana

Spróbuj sterowników binarnych. Dla sterowników ati, listę znajdziesz tu: http://www.schneider-digital.de/html/download_ati.php. Jeżeli to nie pomoże, użyj fbdev. Jest powolny, ale działa.

Mam kartę PCI i nic nie działa. Pomocy!

W sekcji "Device" uruchom ForcePCIMode.

Listing 7.1: Enabling ForcePCIMode

```
Option "ForcePCIMode" "True"
```

9. Kodowanie UTF-8 w Gentoo

1. Kodowanie znaków

Czym jest kodowanie znaków?

Komputery nie rozumieją tekstu wprost. Zamiast tego każdy znak reprezentowany jest przez liczbę. Tradycyjnie każdy zestaw liczb wykorzystywany do reprezentacji alfabetów oraz znaków (znany jako system kodowania (ang. coding system), kodowanie (encoding) lub zestaw znaków (character set) był ograniczony w rozmiarze w wyniku ograniczeń sprzętowych.

Historia kodowania znaków

Najpopularniejszym (a w każdym bądź razie najpowszechniej przyjętym) zestawem jest **ASCII** (American Standard Code for Information Interchange - Amerykański Standardowy Kod dla Wymiany Informacji). Powszechnie uważa się, że ASCII jest najskuteczniejszym standardem w oprogramowaniu. Współczesny ASCII został przyjęty w roku 1986 (ANSI X3.4, RFC 20, ISO/IEC 646:1991, ECMA-6) przez Amerykański Narodowy Instytut Standardów (ang. American National Standards Institute).

ASCII jest kodowaniem ściśle siedmiobitowym - to znaczy wykorzystującym liczby, które da się zapisać na siedmiu bitach, co daje zakres od 0 do 127 (dziesiątkowo). Wśród tych 128 znaków zawarte jest 32 znaki niedrukowalne, w większości zawarte między 0 i 31, oraz znak DEL (czyli skasuj, ang. delete) na pozycji 127. Znaki od 32 do 126 to znaki drukowalne: spacja, znaki przestankowe, litery łacińskie oraz cyfry.

Pierwotnie ósmy znak ASCII wykorzystywany był jako bit parzystości dla kontroli błędów. Jeśli nie było to potrzebne, pozostawiano go jako 0. To znaczy, że w ASCII każdy znak reprezentowany jest przez dokładnie jeden bajt.

Chociaż ASCII było wystarczające dla komunikacji w nowoczesnym angielskim, w innych językach europejskich, w których występowały znaki akcentowane, nie było to już takie proste. Standardy ISO 8859 zostały opracowane, aby sprostać temu problemowi. Były one zgodne z ASCII, ale zamiast pozostawiać ostatni bit pustym, wykorzystywały go, aby dopuścić dodatkowe 127 znaków w każdym kodowaniu. Niestety ograniczenia ISO 8859 stały się szybko widoczne i w chwili obecnej istnieje 15 wariantów tego standardu (od 8859-1 do 8859-15). Poza zakresem zgodnym z ASCII często występują konflikty pomiędzy znakami reprezentowanymi przez każdy bajt. Dodatkową komplikacją jest fakt, że niektóre wersje systemu Microsoft Windows wykorzystują standard Windows-1252 (dla wersji polskiej - Windows-1250). Jest to nadzbiór ISO 8859-1 (Windows-1250 jest podobny do ISO 8859-2), jest jednak na parę sposobów odmienny. Wszystkie te zbiory zachowują jednak zgodność z ASCII.

Niezbędne okazało się też wprowadzenie całkowicie odmiennych kodowań dla alfabetów innych niż łacińskie, takich jak EUC (Extended Unix Coding - Rozszerzone Kodowanie Uniksowe), który jest wykorzystywane w językach Japońskim i Koreańskim (oraz, w ograniczonym zakresie, w Chińskim) - co wprowadziło dodatkowe zamieszanie. Tymczasem inne systemy operacyjne wykorzystywały jeszcze inne kodowania dla tych samych języków, jak na przykład Shift-JIS czy ISO-2022-JP. Użytkownicy chcący oglądać znaki Cyrylicy musieli wybierać pomiędzy KOI8-R dla Rosyjskiego i Bułgarskiego, KOI8-U dla Ukraińskiego oraz innych kodowań Cyrylicy, takich jak nieudany ISO 8859-5 oraz popularny Windows-1251. Wszystkie te kodowania łamały kompatybilność z ASCII (aczkolwiek kodowania KOI8 ustawiają znaki cyrylicy w kolejności Łacińskiej, więc po odcięciu ósmego bitu tekst da się odczytać przez tzw. case-reversed transliteration).

Powstało w ten sposób zamieszanie niemal całkowicie uniemożliwiające komunikację wielojęzykową - szczególnie pomiędzy odmiennymi alfabetami. Problem ten rozwiązać ma Unikod.

Czym jest Unikod?

Unikod odrzuca tradycyjne ograniczenie kodowań jednobajtowych. Wykorzystuje 17 "płaszczyzn" po 65 536 znaków do opisu maksymalnie 1 114 112 znaków. Ponieważ pierwsza płaszczyzna, znana jako "Podstawowa Płaszczyzna Wielojęzykowa" (ang. BMP - Basic Multilingual Plane) zawiera niemal wszystko, co typowy użytkownik kiedykolwiek wykorzysta, wiele osób przyjmuje fałszywe założenie, że Unikod jest zbiorem 16-bitowym.

Unikod jest mapowany na wiele różnych sposobów, ale dwa najpopularniejsze to **UTF** (Unicode Transformation Format - Format Transformacji Unikodu) oraz **UCS** (Universal Character Set - Uniwersalny Zestaw Znaków). Numer za UTF oznacza liczbę bitów w jednej jednostce, natomiast numer za UCS oznacza liczbę bajtów. UTF-8 stał się najpopularniejszą formą wymiany tekstu w Unikodzie dzięki swojej zgodnej z kodowaniami 8-bitowymi naturze. Dlatego właśnie jemu poświęcony jest ten dokument.

UTF-8

UTF-8 jest kodowaniem o zmiennej długości znaku - to znaczy że każdy znak może być opisany przez od jednego do czterech bajtów. Tak więc pierwszy bajt UTF-8 wykorzystywany jest do kodowania ASCII, dzięki czemu jest on w pełni zgodny z ASCII. Znaki ASCII oraz Łacińskie daje się kodować z niewielką nadmiarowością, ponieważ wykorzystany jest tylko pierwszy bit. Użytkownicy alfabetów wschodnich, takich jak japoński, którym przyznano wyższy zakres są niezadowoleni, ponieważ nadmiarowość osiąga nawet 50% dla ich danych.

Jakie korzyści daje UTF-8?

UTF-8 pozwala na pracę w zgodnym ze standardami i akceptowanym międzynarodowo środowisku wielojęzycznym przy stosunkowo niewielkiej nadmiarowości. UTF-8 jest najlepszym sposobem przekazywania znaków spoza ASCII przez Internet - dla poczty, IRC lub prawie każdego innego medium. Pomimo tego wiele osób uważa wykorzystanie UTF-8 w komunikacji on-line za nadużycie. Dobrze jest być świadomym nastawienia społeczności danego kanału, listy mailingowej, czy grupy usenetowej przed wykorzystaniem niezgodnej z ASCII części UTF-8.

2. Konfiguracja UTF-8 w Gentoo Linux

Znajdowanie lub tworzenie lokalizacji UTF-8

Teraz, gdy znamy już zasadę działania Unikodu, możemy przystąpić do wykorzystania go w systemie.

Wstępny wymaganiem dla UTF-8 jest wersja glibc ze wsparciem dla języków narodowych (ang. NLS, National Language Support). Zalecanym sposobem uzyskania tego jest plik /etc/locale.gen i wpisanie polecenia locale-gen. Wykorzystanie tego pliku jest poza zakresem tego dokumentu. Jest opisany w tekście zatytułowanym [Lokalizacja Gentoo Linux](#).

Następnie należy określić czy lokalizacja UTF-8 jest już dostępna dla wybranego języka czy też należy ją utworzyć.

Listing 2.1: Sprawdzanie dostępności lokalizacji UTF-8

```
("pl_PL" należy zastąpić nazwą oczekiwanej lokalizacji)
# locale -a | grep 'pl_PL'
pl_PL
pl_PL.UTF-8
```

Z wyjścia tego polecenia należy wybrać wynik z końcówką .UTF-8. Jeśli żaden z wierszy nie ma takiej końcówki to musimy utworzyć lokalizację zgodną z UTF-8.

Uwaga: Poniższe polecenie należy wywołać tylko wtedy, gdy nie jest dostępna lokalizacja zgodna z UTF-8 dla wybranego języka.

Listing 2.2: Tworzenie lokalizacji UTF-8

```
("pl_PL" należy zamienić na oczekiwana lokalizację)
# localedef -i pl_PL -f UTF-8 pl_PL.UTF-8
```

Innym sposobem uzyskania lokalizacji UTF-8 jest dodanie jej do pliku /etc/locale.gen.

Listing 2.3: Wiersz w /etc/locale.gen

```
pl_PL.UTF-8 UTF-8
```

Wybór lokalizacji

Zmienna środowiskowa LC_ALL musi być ustawiona, aby wybrać lokalizację UTF-8 więcej przy lokalizacjach.(ustawienie tej zmiennej powoduje ignorowanie LANG). Można to zrobić na wiele sposobów; niektórzy preferują kiedy UTF-8 działa tylko dla konkretnego użytkownika - w takim wypadku należy ją ustawić w pliku ~/.profile (jeżeli używa /bin/sh), ~/.bash_profile lub ~/.bashrc (jeżeli używa /bin/bash).

Inni wolą rozwiązanie globalne. Konkretnym przypadkiem w którym autor sugeruje takie rozwiązanie jest sytuacja, gdy wykorzystany jest /etc/init.d/xdm, ponieważ skrypt ten uruchamia menedżer logowania oraz pulpit zanim którykolwiek z plików zostanie odczytany, a tym samym zanim którakolwiek ze zmiennych w nim wymienionych znajdzie się w środowisku.

Globalną lokalizację powinno się konfigurować zatem użyciem pliku /etc/env.d/02locale. Plik ten powinien wyglądać podobnie do tego:

Listing 2.4: Przykładowy /etc/env.d/02locale

```
(Jak zwykle, "pl_PL.UTF-8" zamieniamy na właściwą lokalizację, jeśli ma być inna niż polska)
LC_ALL="pl_PL.UTF-8"
```

Następnie środowisko musi zostać zaktualizowane.

Listing 2.5: Aktualizacja środowiska

```
# env-update
>>> Regenerating /etc/ld.so.cache...
 * Caching service dependencies ...
# source /etc/profile
```

Teraz proszę uruchomić locale bez żadnych parametrów aby sprawdzić, czy właściwe zmienne znajdują się w środowisku:

Listing 2.6: Sprawdzanie, czy nowa lokalizacja znajduje się w środowisku

```
# locale
LANG=
LC_CTYPE="pl_PL.UTF-8"
LC_NUMERIC="pl_PL.UTF-8"
LC_TIME="pl_PL.UTF-8"
LC_COLLATE="pl_PL.UTF-8"
LC_MONETARY="pl_PL.UTF-8"
LC_MESSAGES="pl_PL.UTF-8"
LC_PAPER="pl_PL.UTF-8"
LC_NAME="pl_PL.UTF-8"
LC_ADDRESS="pl_PL.UTF-8"
LC_TELEPHONE="pl_PL.UTF-8"
LC_MEASUREMENT="pl_PL.UTF-8"
LC_IDENTIFICATION="pl_PL.UTF-8"
LC_ALL=pl_PL.UTF-8
```

To wszystko. Lokalizacje UTF-8 jest już w użyciu, następnym problemem jest skonfigurowanie potrzebnych aplikacji.

3. Wsparcie w aplikacjach

Kiedy Unikod zaczynał zdobywać popularność kodowania wielobitowe nie były dobrze wspierane przez języki programowania takie jak C, w których napisane jest wiele popularnych programów. Nawet teraz niektóre programy nie potrafią prawidłowo obsługiwać UTF-8. Na szczęście większość potrafi!

Nazwy plików, NTFS oraz FAT

W konfiguracji jądra jest kilka opcji NLS i istotnym jest, aby się nie pogubić. Najważniejszym jest, aby pamiętać o wbudowaniu NLS UTF-8 w jądro i skonfigurowaniu domyślnego NLS na utf8.

Listing 3.1: Konfiguracja UTF-8 w jądrze

```
File Systems -->
  Native Language Support -->
    (utf8) Default NLS Option
    <*> NLS UTF8
    (<*> również dla innych zestawów znaków które będą
     wykorzystywane na systemach plików FAT lub CD-ROM-ach zgodnych z Joliet).
```

Jeśli planowane jest montowanie partycji NTFS, koniecznym może okazać się określenie opcji nls= podczas montowania. Przy montowaniu partycji FAT potrzebne może okazać się określenie opcji codepage=. Opcjonalnie można ustawić domyślną stronę kodową dla FAT w konfiguracji jądra. Opcja codepage przy montowaniu jest bardziej znacząca od ustawienia w jądrze.

Listing 3.2: Ustawienia FAT w konfiguracji jądra

```
File Systems -->
  DOS/FAT/NT Filesystems -->
    (437) Default codepage for fat
```

Nie należy ustawiać opcji Default iocharset for fat na UTF-8, ponieważ może to sprawiać problemy. W zamian lepiej jest przekazywać do mount opcję utf8=true przy montowaniu partycji FAT. Dalsze informacje można znaleźć w man mount oraz w dokumentacji jądra w /usr/src/linux/Documentation/filesystems/vfat.txt.

Do zmiany kodowania nazw plików można wykorzystać app-text/convvmv.

Listing 3.3: Przykładowe wykorzystanie convvmv

```
# emerge --ask app-text/convvmv
(Składnia polecenia)
# convvmv -f <aktualne-kodowanie> -t utf-8 <nazwa-pliku>
(iso-8859-2 należy zastąpić oryginalnym kodowaniem)
# convvmv -f iso-8859-2 -t utf-8 nazwa-pliku
```

Do zmiany zawartości plików można użyć narzędzia iconv, dostarczanego razem z glibc:

Listing 3.4: Przykładowe wykorzystanie iconv

```
(iso-8859-2 należy zastąpić oryginalnym kodowaniem)
(Sprawdzenie, czy wyjście jest prawidłowe)
# iconv -f iso-8859-2 -t utf-8 nazwa-pliku
(Konwersja pliku, trzeba utworzyć inny plik)
# iconv -f iso-8859-2 -t utf-8 nazwa-pliku > nowy-plik
```

app-text/recode też może być wykorzystany do tego celu.

Konsola systemowa

Ważne: Do obsługi Unikodu na konsoli potrzebny jest sys-apps/baselayout-1.11.9 lub nowszy.

Aby wykorzystać UTF-8 na konsoli należy wyedytować /etc/rc.conf i ustawić UNICODE="yes" oraz przeczytać komentarze w tym pliku. Wykorzystanie fontu z właściwym zakresem znaków jest istotne jeżeli chce się wykorzystać jak najwięcej z możliwości Unikodu.

Zmienna KEYMAP, ustawiana w /etc/conf.d/keymaps, powinna wybierać mapę klawiatury zgodną z Unikodem.

Listing 3.5: Przykładowy fragment /etc/conf.d/keymaps

```
("pl2" należy zamielić na inny układ - jeśli ma być inny niż polski)
KEYMAP="pl2"
```

Ncurses oraz Slang

Uwaga: Jeśli nie korzysta się ze Slang to można zignorować tę część tekstu.

Rozsądnie jest dodać unicode do globalnych flag USE w /etc/make.conf, a następnie ponownie zainstalować sys-libs/ncurses oraz sys-libs/slick.

Listing 3.6: Emergowanie ncurses oraz slang

```
# emerge --update --deep --newuse world
```

Po zmianie flag USE należy też przebudować pakiety które są z nimi zlinkowane. Narzędzie które wykorzystamy (revdep-rebuild) jest częścią pakietu gentoolkit.

Listing 3.7: Przebudowanie programów zlinkowanych z ncurses lub slang

```
# revdep-rebuild --soname libncurses.so.5  
# revdep-rebuild --soname libslang.so.1
```

KDE, GNOME oraz Xfce

Wszystkie popularne środowiska użytkownika mają pełne wsparcie dla Unikodu i nie będą wymagały dalszej konfiguracji poza tym, co już zostało opisane. Wynika to z tego, że biblioteki graficzne które są ich podstawą (Qt oraz GTK+2) mają wsparcie dla UTF-8. W rezultacie wszystkie aplikacje bazujące na tych bibliotekach powinny być zgodne z UTF-8.

Wyjątkami od tej zasady są biblioteki Xlib oraz GTK+. GTK+ wymaga ustawienia FontSpec iso-10646-1 w pliku `~/.glrc`, na przykład `-misc-fixed-*-*-*-*-*-*-*-*iso10646-1`. Aplikacje wykorzystujące Xlib lub Xaw również wymagają podobnego FontSpec, w przeciwnym wypadku będą działały nieprawidłowo.

Uwaga: Jeśli dostępne jest centrum sterowania gnome1 to należy w zamian skorzystać z niego. Wystarczy wybrać w nim dowolny font iso10646-1.

Listing 3.8: Przykładowy `~/.gtkrc` (dla GTK+1) który wybiera font zgodny z Unikodem

Jeżeli aplikacja ma wsparcie dla GUI opartego zarówno na QT jak i GTK+2, to wersja GTK+2 z reguły daje lepsze rezultaty dla Unikodu.

X11 oraz fonty

Ważne: x11-base/xorg-x11 ma znacznie lepsze wsparcie dla Unikodu niż Xfree86 i jest bardzo zalecane.

Fonty TrueType mają wsparcie dla Unikodu, podobnie większość fontów rozpuszczanych razem z Xorg ma bardzo obszerne (choć nadal niekompletne) zbiory znaków. Aby zbudować fonty (włączając w to zestaw Bitstream Vera) ze wsparciem dla znaków Wschodnio-Azjatyckich w X należy upewnić się, że ustawiona jest flaga USE cjk. Wiele innych aplikacji wykorzystuje tę flagę więc w zasadzie warto ją ustawić globalnie.

Również część pakietów fontów dostępnych w Portage ma wsparcie dla Unikodu

Listing 3.9: Opcjonalnie: instalacja dodatkowych fontów wspierających Unikod

```
# emerge terminus-font intlfonts freefonts cronvxs-fonts corefonts
```

Menedżery okien oraz emulatory terminali

Menedżery okien, które nie są oparte na GTK lub Qt z reguły mają bardzo dobre wsparcie dla Unikodu gdyż często wykorzystują do obsługi czcionek bibliotekę Xft. Jeżeli wybrany menedżer okien nie wykorzystuje Xft to nadal można wykorzystać wspomniane wcześniej ustawienie FontSpec.

Emulatory terminala, które wykorzystują Xft i wspierają Unikod są trudniejsze do znalezienia. Poza Konsole i gnome-terminal, najlepszymi wariantami dostępnymi w Portage są x11-terms/rxvt-unicode, xfce-extra/terminal, gnustep-apps/terminal, x11-terms/mlterm, lub zwykły x11-terms/xterm, kiedy zostanie zbudowany z flagą USE unicode oraz wywołany jako uxterm. app-misc/screen również wspiera UTF-8 o ile zostanie wywołany jako screen -U oraz w `~/.screenrc` znajduje się co następuje:

Listing 3.10: `~/.screenrc` dla UTF-8

defutf8_on

Vim, Emacs, Xemacs oraz Nano

Vim ma pełne wsparcie UTF-8 oraz wbudowane rozpoznanie zakodowanych w nim plików. Dalsze informacje można znaleźć po wykonaniu w Vimie polecenia :help mbyte.txt.

Emacs 22.x oraz nowsze również mają pełne wsparcie dla UTF-8. Xemacs 22.x nie ma jeszcze wsparcia dla znaków kombinowanych.

Starsze wersje Emacsa oraz Xemacsa mogą wymagać instalacji app-emacs/mule-ucs oraz app-xemacs/mule-ucs oraz modyfikacji ~/.emacs dla uzyskania wsparcia języków CJK w UTF-8:

Listing 3.11: Emacs ze wsparciem CJK UTF-8

```
(require 'un-define)
(require 'jisx0213)
(set-language-environment "Japanese")
(set-default-coding-systems 'utf-8)
(set-terminal-coding-system 'utf-8)
```

Nano posiada obsługę UTF-8 od wersji 1.3.6.

Powłoki

Już teraz bash zapewnia pełne wsparcie Unikodu za pośrednictwem biblioteki GNU readline. Użytkownicy powłoki Z są w gorszej sytuacji -- powłoka ta wcale nie obsługuje Unikodu, aczkolwiek podejmowane są zdecydowane próby zapewnienia wsparcia dla wielobajtowych zestawów znaków.

Powłoki C, tcsh oraz ksh nie zapewniają żadnego wsparcia dla UTF-8.

Irssi

Irssi ma pełne wsparcie dla UTF-8, wymaga tylko ustawienia opcji przez użytkownika.

Listing 3.12: Uruchamianie UTF-8 w Irssi

```
/set term_charset UTF-8
```

Na kanałach w których znaki spoza ASCII nie są wymieniane w UTF-8 polecenie /recode może zostać użyte do konwersji znaków. Więcej informacji będzie dostępne po wpisaniu polecenia /help recode.

Mutt

Klient poczty Mutt ma bardzo dobре wsparcie dla Unikodu. Aby z niego skorzystać wystarczy zakodować wszystkie pliki konfiguracyjne (łącznie z sygnaturką) w UTF-8.

Uwaga: Nadal może się zdarzyć, że w odebranej przez Mutta poczcie znajdzie się znak "?" zamiast właściwego. Wynika to z tego, że niektórzy korzystają z klientów które nie oznaczają wykorzystanego kodowania. Nie da się z tym wiele zrobić, poza poproszeniem ich o prawidłową konfigurację klienta.

Dalsze informacje są dostępne na [Wiki Mutta](#).

Man

Strony man są istotną częścią każdego komputera z Linuksem. Aby zapewnić prawidłowe wyświetlanie znaków Unikodu należy zmodyfikować /etc/man.conf:

Listing 3.13: Modyfikacja man.conf dla wsparcia Unikodu

```
(To jest stary wiersz)
NROFF          /usr/bin/nroff -Tascii -c -mandoc
(Powyższe należy zastąpić poniższym)
NROFF          /usr/bin/nroff -mandoc -c
```

elinks oraz links

Są to powszechnie wykorzystywane przeglądarki tekstowe i spróbujemy uruchomić w nich wsparcie dla UTF-8. Dla elinks oraz links są na to dwa sposoby, jeden z nich polega na wykorzystaniu opcji Setup z wewnątrz przeglądarki, drugi zaś na edycji pliku konfiguracyjnego. Aby ustawić opcję z wewnątrz przeglądarki należy uruchomić elinks lub links a następnie wykorzystać Alt+S, aby wejść do menu Setup, następnie wybrać pozycję Terminal options lub nacisnąć klawisz T. Teraz należy przewinąć w dół i zaznaczyć ostatnią opcję: UTF-8 I/O. Następnie należy zapisać i opuścić menu. Pod links może być potrzebne ponowne naciśnięcie Alt+S oraz naciśnięcie S aby zapisać. Opcja w pliku konfiguracyjnym pokazana jest poniżej.

Listing 3.14: Uruchamianie UTF-8 dla elinks/links

```
(Dla elinksa należy wyedytować /etc/elinks/elinks.conf lub
~/.elinks/elinks.conf oraz dodać następujący wiersz)
set terminal.linux.utf_8_io = 1

(Dla linksa należy wyedytować ~/.links/links.cfg oraz dodać następujący wiersz)
terminal "xterm" 0 1 0 us-ascii utf-8
```

Samba

Samba to otwarta implementacja protokołu SMB (Server Message Block) dla systemów uniksowych takich jak Linux, Mac i FreeBSD. Protokół ten czasem jest nazywany CIFS (Common Internet File System). Częścią Samby jest również system NetBIOS używany do współdzielenia plików z systemami Windows.

Listing 3.15: Włączanie UTF-8 dla Samby

```
(Edytujemy plik /etc/samba/smb.conf i dodajemy poniższe wpisy do sekcji [global])
dos charset = 1255
unix charset = UTF-8
display charset = UTF-8
```

Testowanie całości

Jest wiele stron poświęconych testowaniu UTF-8. net-www/w3m, net-www/links, net-www/elinks, net-www/lynx oraz wszystkie przeglądarki oparte na Mozilli (włącznie z Firefoksem) wspierają UTF-8. Konqueror oraz Opera również mają pełne wsparcie UTF-8.

Korzystając z przeglądarek tekstowych należy upewnić się, że terminal obsługuje Unikod.

Jeżeli zamiast niektórych znaków widoczne są kwadraty z literami oraz cyframi w środku to oznacza to, że font nie zawiera danego znaku UTF-8. Zamiast tego wyświetla pudełko z jego kodem szesnastkowym.

- [Strona testowa UTF-8 na W3C](#)
- [Strona testowa UTF-8 dostarczona przez Uniwersytet we Frankfurcie](#)

Metody wejściowe

Martwe klawisze mogą być wykorzystane do wpisywania w X znaków, których nie ma na klawiaturze. Działają one przez wcisknięcia prawego klawisza Alt (lub, w niektórych krajach, AltGr) oraz opcjonalnie klawisza z niealfabetycznej części klawiatury na lewo od klawisza return, puszczenia ich, a następnie naciśnięcia znaku. Martwy klawisz powinien go zmodyfikować. Dalsze modyfikacje są możliwe przez użycie klawisza Shift w tym samym czasie, co AltGr oraz modyfikatora.

Aby wykorzystać martwe klawisze w X potrzebny jest układ klawiatury, który to wspiera. Większość układów europejskich zawiera martwe klawisze w domyślnym wariantie. Niestety nie dotyczy to układów północnoamerykańskich. Pomimo pewnego stopnia niekonsekwencji pomiędzy układami, najbliższym rozwiązaniem wydaje się wykorzystania układu w postaci "pl_PL" zamiast "pl". Układ wybierany jest w /etc/X11/xorg.conf w następujący sposób:

Listing 3.16: Fragment /etc/X11/xorg.conf

```
Section "InputDevice"
    Identifier "Keyboard0"
    Driver      "kbd"
    Option     "XkbLayout" "pl_PL" # Zamiast zwykłego "pl"
    (Tu inne opcje Xkb)
EndSection
```

Uwaga: Ta zmiana potrzebna tylko jeśli używane są układy północnoamerykańskie lub inny uład w którym nie działają martwe klawisze. Użytkownicy europejscy nie powinni mieć problemu z martwymi klawiszami.

Ta zmiana zacznie działać dopiero po ponownym uruchomieniu serwera X. Aby zastosować te zmiany od razu należy skorzystać z narzędzia setxkbmap, na przykład setxkbmap pl_PL.

Najprościej jest opisać martwe klawisze za pomocą przykładów. Chociaż rezultaty są zależne od lokalizacji idea pozostaje zawsze taka sama. Te przykłady korzystają z UTF-8, więc aby z nich skorzystać należy albo skonfigurować przeglądarkę do pokazywania strony jako UTF-8, albo wykorzystać lokalizację z UTF-8.

Kiedy naciśnie się AltGr oraz [jednocześnie, puści je, a następnie naciśnie a, wyświetlane zostaje 'ä'. Kiedy naciśnie się AltGr oraz [, następnie e, wyświetlane zostaje 'ë'. Kiedy naciśnie się AltGr oraz ;, wyświetlane zostaje 'á'. Kiedy naciśnie się AltGr oraz ; a następnie e wyświetlone zostanie 'é'.

Kiedy naciśnie się jednocześnie AltGr, Shift oraz [, puści się je, a następnie naciśnie a, wyświetlane zostanie Skandynawskie 'å'. Podobnie, po naciśnięciu altGr, Shift oraz [,пусzczeniu wyłącznie [, następnie naciśnięciu go ponownie, wyświetlane zostanie 'ö'. Wprawdzie znak ten (U+02DA) wygląda podobnie, ale nie jest on tym samym co symbol stopnia (U+00B0). Działa to również dla innych akcentów produkowanych przez martwe klawisze - AltGr oraz [, puszczenie tylko [oraz naciśnięcie go ponownie da ""/>.

AltGr może też zostać użyty z samym klawiszem alfabetycznym. Na przykład, AltGr i m dadzą w rezultacie małą literę mu: 'μ'. AltGr oraz s dadzą scharfes s: 'ß'. Zgodnie z oczekiwaniemi AltGr oraz 4 daje symbol Euro, '€'.

10. Lokalizacja Gentoo Linux

1. Strefa czasowa

Aby czas był ustawiany poprawnie, dowiązanie /etc/localtime musi być plikiem prawidłowej strefy czasowej. Należy przeszukać katalog /usr/share/zoneinfo/ i odnaleźć odpowiedni plik dla kraju (kontynentu) i najbliższego dużego miasta. Należy unikać stref czasowych o nazwie /usr/share/zoneinfo/Etc/GMT*, ponieważ ich nazwy mogą być mylące, na przykład GMT-8 jest w rzeczywistości GMT8.

Listing 1.1: Konfiguracja strefy czasowej

```
# cp /usr/share/zoneinfo/Europe/Warsaw /etc/localtime
# date
pią sie 19 03:09:58 CEST 2005
```

Uwaga: Należy sprawdzić czy trzy lub cztery litery kodu strefy czasowej są poprawne (w naszym polskim przypadku będą to "CEST" (Central European Summer Time) latem lub "CET" (Central European Time) zimą).

Uwaga: Można ustawić wartość TZ (również wziętą z /usr/share/zoneinfo w pliku rc (.bash_profile dla basha), aby skonfigurować strefę czasową tylko dla jednego użytkownika. W naszym przypadku może to być TZ="Europe/Warsaw".

2. Zegar sprzętowy

W większości systemów Gentoo Linux, zegar sprzętowy jest ustawiony na UTC (lub GMT, Greenwich Mean Time), a następnie strefa czasowa jest brana pod uwagę przy ustalaniu konkretnej godziny. Jeżeli z jakichś powodów zegar sprzętowy nie jest ustawiony na UTC, to należy wyedytować /etc/rc.conf i zmienić wartość zmiennej CLOCK.

Listing 2.1: Zegar lokalny vs. GMT

```
Rekomendowany:
CLOCK="UTC"
Alternatywny:
CLOCK="local"
```

3. Lokalizacje systemu

Co to jest lokalizacja?

Lokalizacje są zbiorem informacji dla większości programów pozwalającym na ustalenie kraju i języka. Lokalizacje i informacje o nich są częścią biblioteki systemowej i możesz je znaleźć w /usr/share/locale w większości systemów. Nazwy lokalizacji mają postać ab_CD, gdzie ab jest kodem języka (jak określono w ISO-639), a CD kodem kraju (z ISO-3166).

Zmienne środowiskowe dla lokalizacji

Ustawienia lokalizacji są zapisywane w zmiennych środowiskowych, które ustawia się w pliku /etc/env.d/02locale (dla ustawień całego systemu) i pliku ~/.bashrc (dla użytkownika). Zmienne kontrolują różne aspekty ustawień lokalizacji, omawiamy je w tabeli poniżej, te z większym priorytetem są na górze tabelki. Ustawienia o większym priorytecie nadpiszą te z priorytetem mniejszym.

Wszystkie zmienne mają jeden format lokalizacji ab_CD - tak jak opisano to powyżej.

Nazwa zmiennej Opis

LC_ALL	Definiuje wszystkie ustawienia w lokalizacji naraz. Jest to najwyższy poziom ustawień dla lokalizacji i zastąpi wszystkie inne ustawienia.
LC_COLLATE	Definiuje porządek alfabetyczny. Może mieć efekt na np. wyświetlenie zawartości katalogów.
LC_CTYPE	Definiuje zarządzanie ustawieniami znaków dla systemu. Np. które znaki są widziane jako część alfabetu, a które jako numeryczne. Czasami ustawia to również zestaw używanych znaków.
LC_MESSAGE	Ustawienie dla aplikacji, które używają zmiennych lokalizacji do wyświetlania wiadomości (większość programów GNU, więcej informacji w następnym rozdziale).
LC_MONETARY	Definiuje jednostkę waluty i jej format numeryczny.
LC_NUMERIC	Definiuje formatowanie wartości numerycznych które nie są walutą. Ustala np. separatory tysiący i dziesiątek.
LC_TIME	Definiuje format daty i czasu.
LC_PAPER	Definiuje domyślny rozmiar papieru.
LANG	Definiuje wszystkie lokalne ustawienia. Te ustawienia mogą być nadpisane przez indywidualne ustawienia LC_* powyżej lub nawet przez LC_ALL.

Uwaga: Mimo, że większość programów współpracuje jedynie z LC_ALL warto również ustawić zmienną LANG, bez której część z nich może się dziwnie zachowywać.

Większość użytkowników ustawia zmienne LANG i LC_CTYPE na poziomie użytkownika przez ręczne dodanie definicji zmiennych do skryptów startowych powłok.

Listing 3.1: Ustawianie polskiej lokalizacji

```
export LANG="pl_PL"
```

Uwaga: Dodajemy @euro do lokalizacji jeżeli chcemy używać symbolu tej waluty (€). Nie ma go w polskich lokalizacjach, ponieważ nie znajduje się w tabeli znaków normy iso8859-2.

Możliwe jest skonfigurowanie systemu tak, aby używał lokalizacji dla wszystkich użytkowników i programów.

Listing 3.2: Konfiguracja lokalizacji w pliku /etc/env.d/02locale

```
LC_ALL="pl_PL"
LANG="pl_PL"
```

Częstym zwyczajem jest pozostawienie systemowych ustawień lokalizacji bez zmian i konfiguracja ich jedynie dla poszczególnych użytkowników. W takim wypadku system będzie korzystał z domyślnego ustawienia "C", które z powodów historycznych jest angielskie.

Aby wiadomości w programach działały jak należy, konieczne będzie skompilowanie ich z obsługą nl (native language support), którą zapewnia flaga USE o tej samej nazwie. W takim wypadku do wyświetlania właściwych wiadomości konieczna będzie biblioteka gettext. Portage zainstaluje ją automatycznie kiedy tylko będzie potrzebna.

Gdy mamy już ustawione odpowiednie lokale, należy upewnić się, że zaaktualizowaliśmy zmienną środowiskową:

Listing 3.3: Aktualizacja środowiska

```
(Gdy lokalizacje zostały ustawione dla całego systemu)
# env-update && source /etc/profile
```

```
(Gdy lokalizacje zostały ustawione tylko dla poszczególnych użytkowników)
$ source ~/.bashrc
```

Po aktualizacji, należy zabić serwer X poprzed kombinację klawiszy Ctrl-Alt-Backspace, wylogować się, a następnie zalogować jako użytkownik.

Powinniśmy sprawdzić czy zmiany, które wprowadziliśmy przyniosły oczekiwane działanie:

Listing 3.4: Weryfikacja zmian

```
$ env | grep -i LC_
```

Istnieje dodatkowa zmienna mająca wpływ na instalowane lokalizacje, ma ona nazwę LINGUAS. Pozwala ona na określenie plików językowych jakie mają być instalowane wraz z aplikacjami opartymi na gettext. Pozwala ona również na wybranie i zainstalowanie tylko niektórych z lokalizacji dostępnych w pakietach, takich jak kde-base/kde-i18n, czy app-office/openoffice. Ustawia się ją w pliku /etc/make.conf i ma ona postać listy kodów języków, jakie chcemy zainstalować oddzielonych spacjami.

Listing 3.5: Ustawianie zmiennej LINGUAS w pliku make.conf

```
# nano -w /etc/make.conf
(Dodajemy zmienną LINGUAS. Przykład dla języków angielskiego, fińskiego i polskiego:)
LINGUAS="en fi pl"
```

Generowanie specyficznych lokalizacji

Do wygenerowania lokalizacji innych niż domyślne służy program localedef. Na przykład:

Listing 3.6: Generowanie lokalizacji za pomocą localedef

```
# localedef -c -i pl_PL -f ISO-8859-2 pl_PL.ISO-8859-2
```

Po wygenerowaniu lokalizacji, można je wyeksportować jako zmienną LANG.

Listing 3.7: Eksportowanie zmiennej LANG

```
# export LANG="pl_PL.ISO-8859-2"
```

Należy zaaktualizować środowisko po dokonaniu zmian

Listing 3.8: Aktualizacja środowiska

```
# env-update && source /etc/profile
```

Po aktualizacji, należy zabić serwer X poprzed kombinację klawiszy Ctrl-Alt-Backspace, wylogować się, a następnie zalogować jako użytkownik.

Budowanie lokalizacji dla glibc

W systemie rzadko kiedy używa się więcej niż jednej, góra dwóch lokalizacji. Można je wybrać w pliku /etc/locale.gen.

Listing 3.9: Dodawanie wpisów do pliku /etc/locale.gen

```
en_US ISO-8859-1
en_US.UTF-8 UTF-8
pl_PL ISO-8859-2
```

Kolejnym krokiem jest przekompilowanie i uruchomienie programu locale-gen. Utworzy on wszystkie lokalizacje określone w pliku /etc/locale.gen.

Uwaga: locale-gen jest dostępne w glibc-2.3.6-r4 i nowszych. Jeśli w systemie jest zainstalowana starsza wersja, należy ją zaktualizować.

4. Ustawienia klawiatury dla konsoli

Ustawienie klawiatury używane przez konsole jest konfigurowane w pliku /etc/conf.d/keymaps przez zmienną KEYMAP. Poprawne wartości można znaleźć w /usr/share/keymaps/{arch}/. i386 ma podgromady (qwerty/, azerty/...). Niektóre języki mają kilka opcji, należy na drodze eksperymentów wybrać najlepsze ustawienia.

Listing 4.1: Ustawianie mapowania klawiatury dla konsoli

```
KEYMAP="pl"
```

5. Mapowanie klawiatury dla serwera X

Mapowanie klawiatury używane przez serwer X jest ustawianie w pliku /etc/X11/xorg.conf w zmiennej XkbLayout.

Listing 5.1: Ustawianie mapowania dla X

```
Section "InputDevice"
    Identifier "Keyboard1"
    ...
    Option "XkbLayout"      "pl2"
    #Option "XkbModel"      "pc105"      ## ustawienie dla
międzynarodowych klawiatur.
    # Option "XkbVariant"   "nodeadkeys" ## ustawienie zmiennej
używanej przez xterm
    ...

```

Jeżeli posiadamy międzynarodowy układ klawiatury, powinniśmy ustawić opcję XkbModel na wartość pc102 lub pc105. Pozwoli to mapować dodatkowe klawisze specyficzne dla danej klawiatury.

Opcja martweklawisze (deadkeys) pozwala na wciskanie klawiszy, których znaki nie pojawiają się od razu, a muszą zostać wciśnięte w kombinacji z innymi, aby wyświetlić znak np. é, á itp. Ustawienie opcji XkbVariant na nodeadkeys pozwala wprowadzać znaki specjalne do terminali X.

Jeżeli chcemy przełączać się między więcej niż jednym układem klawiatury (dla przykładu między rosyjskim i angielskim), wystarczy dodać kilka linijek do pliku xorg.conf, aby zdefiniować porządkowane przez nas ustawienia klawiatury oraz skort do przełączania się między nimi.

Listing 5.2: Przełączanie się pomiędzy dwoma układami klawiatury

```
Section "InputDevice"
    Identifier "Keyboard1"
    ...
    Option "XkbLayout"      "us,ru"
    Option "XkbOptions"    "grp:alt_shift_toggle,grp_led:scroll"
```

Ustawienie opcji XkbOptions pozwala na przełączanie się pomiędzy różnymi układami klawiatury poprzez prostą kombinację klawiszy Alt-Shift. Dzięki opcji grp_led:scroll włączana i wyłączana będzie również dioda Scroll Lock. Jest to poręczna informacja, pozwalająca rozpoznać jakiego układu używamy w danym momencie.

6. KDE

W celu spolszczenia KDE należy zainstalować pakiet kde-i18n. Sporo ułatwi ustawienie wcześniej zmiennej LINGUAS, którą opisaliśmy nieco wcześniej.

7. Symbol Euro w konsoli

Aby wyświetlał się symbol Euro, należy ustawić zmienną CONSOLEFONT w /etc/conf.d/consolefont. Wartość jaką należy do niej wpisać można odnaleźć w katalogu /usr/share/consolefonts/. Należy wybrać nazwę czcionki i opuścić rozszerzenie pliku (.psfu.gz). Symbol euro znajduje się na przykład w czcionce lat2a-16.

Listing 7.1: Ustawienie fontów dla konsoli

```
CONSOLEFONT="lat2a-16"
```

Powiniśmy sprawdzić czy CONSOLEFONT znajduje się w poziomie uruchomieniowym boot:

Listing 7.2: Weryfikacja poziomu uruchomieniowego

```
# rc-update show | grep -i consolefont
```

Jeżeli CONSOLEFONT nie znajduje się na żadnym z poziomów uruchomieniowych, należy je na jakiś dodać.

Listing 7.3: Dodawanie consolefont do prawidłowego poziomu uruchomieniowego

```
# rc-update add consolefont boot
```

8. Symbol Euro w X

Większość aplikacji

Zmuszenie X-ów do wyświetlania symbolu Euro jest trochę bardziej skomplikowane. Pierwszą rzeczą, jaką należy zrobić jest zmiana definicji fixed i variable w pliku /usr/X11R6/lib/X11/fonts/misc/fonts.alias tak, by kończyły się ciągiem iso8859-2 zamiast iso8859-1.

Listing 8.1: Ustawienie domyślnych czcionek dla X

```
fixed      -misc-fixed-medium-r-semicondensed--13-120-75-75-c-60-iso8859-2
variable   -*-helvetica-bold-r-normal-*-*-120-*-*-*-iso8859-2
```

Niektóre aplikacje używają własnych czcionek i należy skonfigurować je oddzielnie do używania takich z symbolem Euro. Można to zrobić na poziomie użytkownika w pliku .Xdefaults (plik można skopiować do /etc/skel/, aby stał się domyślnym dla nowych użytkowników) lub na poziomie globalnym dla każdej aplikacji w plikach /usr/X11R6/lib/X11/app-defaults/ (jak xterm). W tych plikach generalnie trzeba zmienić istniejącą linię, niż dodawać nową. Na przykład, aby zmienić czcionkę dla xtermu:

Listing 8.2: Ustawianie czcionek dla xterm

```
(W katalogu domowym)
$ echo 'XTerm*font: fixed' >> .Xresources
$ xrdb -merge .Xresources
```

Symbol Euro w (X)Emacs

Aby używać symbolu Euro w (X)Emacs, należy dodać poniższą linijkę do pliku .Xdefaults:

Listing 8.3: Ustawianie czcionek dla emacs-a

```
Emacs.default.attributeFont: -*-courier-medium-r-*-*-120-*-*-*-iso8859-2
```

Dla XEmacs (nie Emacs), należy dokonać więcej zmian i dodać następujący wpis w pliku /home/user/.xemacs/init.el:

Listing 8.4: Ustawianie czcionek dla xemacs

```
(define-key global-map '(EuroSign) '[\u20ac])
```

Uwaga: Symbol w [] jest symbolem Euro.

OpenOffice.org

Ebuild app-office/openoffice oraz app-office/openoffice-bin korzysta ze zmiennej LINGUAS w celu wybrania odpowiedniego pakietu językowego dla interfejsu użytkownika.

11. Konfiguracja sterowników ALSA w Gentoo

1. Wprowadzenie

Czym jest ALSA?

ALSA (Advanced Linux Sound Architecture) to system zapewniający obsługę audio i MIDI (Musical Instrument Digital Interface) w systemach opartych na jądrze Linux. Począwszy od jąder serii 2.6, ALSA stała się domyślnym systemem dźwiękowym, wypierając nieco już przestarzały OSS (Open Sound System), używany w jądrach 2.4.

Główne zalety systemu ALSA to wydajna obsługa wszystkich typów kart dźwiękowych, poczynając od najtańszego sprzętu, a kończąc na w pełni profesjonalnych rozwiązańach, modularna budowa sterowników, bezpieczeństwo wątków i współpraca z SMP oraz pełna wsteczna kompatybilność z OSS, a także biblioteki alsa-lib znacznie ułatwiające tworzenie nowego oprogramowania.

ALSA w Gentoo

Jedną z głównych zalet Gentoo jest możliwość pełnej kontroli nad procesem instalacji i konfiguracji systemu. Zasada ta dotyczy także systemu ALSA. Poniżej zostały omówione dwie metody instalacji tego systemu obsługi dźwięku.

2. Instalowanie ALSA

Opcje

Ostrzeżenie: Dwie metody opisane poniżej wzajemnie się wykluczają. Nie można posiadać jednocześnie obsługi ALSA wkompilowanej w jądro i zainstalowanej poprzez ebuild media-sound/alsa-driver. To się nie uda.

Możliwości są następujące:

1. Sterowniki ALSA wbudowane w jądro - rozwiązanie zalecane
2. Pakiet media-sound/alsa-driver

Warto pokróćce omówić oba rozwiązania.

Oto plusy i minusy sterowników ALSA wbudowanych w jądro:

ALSA z kernela Plusy i minusy

- | | |
|---|-------------------------------------------------------------------|
| + | Sterowniki wbudowane w jądro są stabilniejsze |
| + | Proste rozwiązanie, jednorazowa instalacja |
| - | Mogą być nieco starsze niż sterowniki z pakietu alsadriver |

Plusy i minusy pakietu alsadriver:

alsa-driver Plusy i minusy

- | | |
|---|---------------------------------------------------------------------------------------|
| + | Najświeższe sterowniki ALSA na rynku |
| - | Każda kolejna zmiana jądra wymaga ponowej instalacji pakietu alsadriver. |
| - | Aby pakiet działał prawidłowo określone opcje w jądrze muszą zostać wyłączone. |

Podsumowując...

Główna różnicą pomiędzy pakietem alsadriver, a sterownikami jądra jest to, że ten pierwszy jest zwykle aktualniejszy. Różnice są jednak na tyle nieznaczne, że raczej nie warto męczyć się ze sterownikami spoza jądra, poza tym instalacja systemu ALSA już w jądrze Linuksa jest na pewno znacznie prostsza. Przed zgłoszeniem do Bugzilli jakichkolwiek błędów związanych ze sterownikami znajdującymi się w jądrze należy sprawdzić, czy błąd występuje także w tych z pakietu alsadriver. Jednak niezależnie od rezultatu tego testu i tak prosimy o zgłaszanie błędów.

Zanim zaczniemy

Obojętnie z której metody korzystamy, konieczna jest znajomość modelu karty dźwiękowej, aby móc wybrać dla niej odpowiedni sterownik. W większości przypadków karty dźwiękowe (nieważne czy zintegrowane na płycie głównej czy nie) to urządzenia PCI, dzięki czemu można skorzystać z programu lspci do poznawania wymaganych o nich informacji. Jeśli nie ma takiego polecenia w systemie to należy wykonać emerge sys-apps/pciutils, aby stało się ono dostępne. Jeśli posiadamy kartę USB to pomocne może być polecenie lsusb z pakietu sys-apps/usbutils. Dla kart ISA warto użyć pakietu sys-apps/isapnptools.

Poniżej znajduje się lista stron, na których można znaleźć informacje o kartach ISA:

- [Oficjalna strona pakietu ISAPNPTOOLS](#)
- [Artykuł z LinuxJournal na temat PnP](#)
- [Przewodnik dotyczący dźwięku pochodzący z projektu dokumentacji Linuksa](#)

Uwaga: Dla uproszczenia tego tekstu zakładamy, że użytkownik posiada kartę PCI.

Pora zatem dowiedzieć się jaką właściwie kartę dźwiękową posiadamy.

Listing 2.1: Szczegóły dotyczące karty dźwiękowej

```
# lspci -v | grep -i audio
0000:00:0a.0 Multimedia audio controller: Creative Labs SB Live! EMU10k1 (rev 06)
```

Teraz wiemy, że karta dźwiękowa w komputerze to Sound Blaster Live! i że jej producentem jest firma Creative Labs. Udajemy się zatem na stronę [ALSA Soundcard Matrix](#) i wybieramy Creative Labs z listy na dole. Zostaniemy przeniesieni na podstronę poświęconą tej firmie, na której będziemy mogli odczytać, że tę kartę obsługuje sterownik emu10k1. I to jest właśnie informacja, której tu szukaliśmy. Jeśli chcemy poznać szczegóły możemy kliknąć na nazwę sterownika i poczytać o nim na jego podstronie.

Używanie sterowników ALSA dostarczanych razem z kernelem

Jest to najlepszy wybór dla osób lubiących prostotę i wygodne rozwiązań.

Uwaga: Od wydania 2005.0 Gentoo korzysta domyślnie z jądra serii 2.6. Na większości architektur, o ile nie wybrano innego profilu, domyślnym jądem są gentoo-sources. Warto sprawdzić czy używane jądro to na pewno 2.6. Metoda, z której będziemy korzystać nie zadziała na 2.4.

Włączmy i skonfigurujmy sterowniki ALSA:

Ważne: Użytkownicy programu genkernel muszą wpisać polecenie genkernel --menuconfig all oraz postępować zgodnie ze wskazówkami z części Sterowniki ALSA w jądrze.

Listing 2.2: Przechodzenie do konfiguracji kernela

```
# cd /usr/src/linux
# make menuconfig
```

Uwaga: Dowiązanie /usr/src/linux musi prowadzić do źródeł kernela, którego zamierzamy używać. Warto sprawdzić czy to prawda przez rozpoczęciem jakichkolwiek operacji opisanych poniżej.

Dokonamy teraz przeglądu dostępnych w jądrach serii 2.6 opcji konfiguracyjnych, zapewniających obsługę określonej karty dźwiękowej.

Warto zwrócić uwagę, że dla uproszczenia we wszystkich przykładach budujemy sterowniki w postaci modułów. Radzimy postąpić podobnie oraz nie zapominać o czynnościach opisanych w akapicie Konfiguracja tego tekstu. Dla sterowników wbudowanych na stałe w jądro trzeba dokonać odpowiedniej korekty w pewnych punktach konfiguracji.

Listing 2.3: Sterowniki ALSA w jądrze

```
Device Drivers --->
  Sound --->

  (To trzeba włączyć)
  <M> Sound card support

  (Tej opcji nie włączamy!)
  Open Sound System --->
    < > Open Sound System (DEPRECATED)
```

```
(Menu wyżej przechodzimy do opcji ALSA)
Advanced Linux Sound Architecture --->
    <M> Advanced Linux Sound Architecture
        (Włącza obsługę MIDI)
    <M> Sequencer support
(Obsługa /dev/mixer* i /dev/dsp* w starym stylu. Polecamy.)
    <M> OSS Mixer API
    <M> OSS PCM (digital audio) API
```

(Teraz należy wybrać urządzenia jakich używamy. W większości przypadków jest to tylko jedna karta. Jeśli jednak jest ich więcej należy zaznaczyć sterowniki dla wszystkich jakie posiadamy)

(Nie jest to konieczne dla zwykłych użytkowników, a przeznaczone głównie dla doświadczonych osób do różnego rodzaju testów)
Generic devices --->

(Dla kart dźwiękowych na slacie ISA)
ISA devices --->
(Dla posiadaczy karty Gravis)
 <M> Gravis UltraSound Extreme

(Wracamy poziom wyżej i przechodzimy do urządzeń PCI. Większość kart dźwiękowych obecnie to właśnie karty PCI)
PCI devices --->
 (Wybieramy sterownik emu10k1 dla naszej karty)
 <M> Emu10k1 (SB Live!, Audigy, E-mu APS)
 (Dla karty Intel)
 <M> Intel/SiS/nVidia/AMD/ALi AC97 Controller
 (Dla karty VIA)
 <M> VIA 82C686A/B, 8233/8235 AC97 Controller

(Poziom wyżej znajduje się podobne menu do wyboru kart USB
USB Devices --->

Po skonfigurowaniu wszystkich opcji należy ponownie zbudować jądro i uruchomić z niego system tak, aby obsługa ALSA stała się dostępna. Po wszystkim należy zaktualizować konfigurację GRUB-a. Kolejny etap konfiguracji to Narzędzia do pracy z systemem ALSA.

Użycie pakietu ze sterownikami ALSA

Przed zainstalowaniem pakietu ze sterownikami (alsa-driver) należy wykonać kilka drobnych czynności. Nie są one wprawdzie wymagane, ale znacznie skrócą czas budowy pakietu poprzez wybranie do komplikacji tylko potrzebnego sterownika, a nie wszystkich, jak to jest domyślnie.

Często nie wiadomo jaki sterownik będzie potrzebny. Wtedy należy po pierwsze zapoznać się z akapitem tego tekstu dotyczącym lspci, a następnie znając już nazwę odpowiedniego sterownika (np. emu10k1), wyeksportować ją w postaci zmiennej ALSA_CARDS w pliku /etc/make.conf.

Listing 2.4: Dodawanie zmiennej ALSA_CARDS do pliku make.conf

```
(Dla jednej karty)
ALSA_CARDS="emu10k1"
(Dla wielu kart - poszczególne nazwy oddzielamy spacjami)
ALSA_CARDS="emu10k1 via82xx"
```

Przed rozpoczęciem instalacji pakietu alsa-driver należy się upewnić, że w konfiguracji jądra uwzględniono poniższe opcje, w przeciwnym wypadku instalacja się nie powiedzie. W przykładach poniżej pokażemy jeden ze sposobów sprawdzenia czy wszystko jest ustawione prawidłowo.

Uwaga: Użytkownicy genkernel powinni przejść do Instalacji alsa-driver, ponieważ ich konfiguracja jest już taka, jak ta przedstawiona poniżej.

1. CONFIG_SOUND musi być włączony (opcja Basic Sound support)
2. CONFIG_SOUND_PRIME jest wyłączony (opcja In-built OSS support)
3. CONFIG_SND jest wyłączony (wyłączona wbudowana obsługa ALSA)
4. /usr/src/linux wskazuje na dobry katalog ze źródłami

Listing 2.5: .config checks

(Zakładając, że dowiązanie o nazwie linux wskazuje na dobry katalog)

```
# cd /usr/src/linux
# grep SOUND .config
(1 to prawda)
CONFIG_SOUND=y
(2 to prawda)
CONFIG_SOUND_PRIME is not set
# grep SND .config
(i 3 to prawda)
CONFIG_SND is not set
```

Na koniec wystarczy wpisać magiczne słowa:

Listing 2.6: Instalowanie pakietu alsa-driver

```
# emerge alsa-driver
```

Ważne: Po każdym ponownym skompilowaniu jądra (co wiąże się ze skasowaniem wszystkich dodatkowych sterowników) należy ponownie wykonać polecenie emerge alsa-driver.

3. Konfigurowanie i testowanie ALSA

Narzędzia do pracy z systemem ALSA

alsa-utils to pakiet zawierający wiele bardzo użytecznych programów oraz skrypty startowe dla ALSA. Jest on integralną częścią tego systemu i dlatego bardzo mocno zachęcamy do jego zainstalowania.

Listing 3.1: Instalowanie alsa-utils

```
# emerge alsa-utils
```

Uwaga: Po odpowiednim skonfigurowaniu jądra i dokompilowaniu dla ALSA odpowiednich modułów pozostaje jeszcze skonfigurować odpowiednie skrypty startowe. Później zajmiemy się dostosowywaniem systemu do naszych potrzeb za pomocą programu alsactl z pakietu alsa-utils.

Konfiguracja

Uwaga: Przed uruchomieniem alsactl należy zamknąć wszystkie programy, które mogą korzystać z karty dźwiękowej w trakcie jej konfiguracji.

Najprostszym programem służącym do konfiguracji karty jest alsactl. Wpisujemy zatem to polecenie z konta roota.

Listing 3.2: Uruchamianie alsactl

```
# alsactl
```

Pojawi się menu, które automatycznie sprawdzi wszystkie urządzenia w komputerze, spróbuje uzyskać informację o karcie dźwiękowej i poprosi o wybranie modelu z listy. Potem zapyta o pozwolenie na automatyczną zmianę pliku /etc/modules.d/alsa, a później przystąpi do dostosowywania poziomów głośności w celu wybrania optymalnych do danej sytuacji. Na koniec uruchomi polecenie modules-update i wystartuje skrypt /etc/init.d/alsasound. Kiedy alsactl zakończy pracę będzie można przystąpić do

konfiguracji skryptu startowego.

Skrypt startowy ALSA

To już niemal wszystko. Obojętnie którą metodę instalowania sterowników wybrano, będzie potrzebne coś co je załaduje oraz odtworzy poziom głośności przy każdym uruchomieniu komputera. Zajmuje się tym skrypt startowy o nazwie alsasound. Należy go dodać do poziomu uruchomieniowego o nazwie "boot".

Listing 3.3: Dodawanie ALSA do poziomu boot

```
# rc-update add alsasound boot
 * alsasound added to runlevel boot
 * rc-update complete.
```

Następnie należy sprawdzić plik /etc/conf.d/alsasound i upewnić się, że zmienna SAVE_ON_STOP ma wartość "yes". Dzięki temu ustawienia głośności będą zapisywane przed wyłączeniem komputera.

Grupa audio

Przed przystąpieniem do testowania nowych ustawień należy ustawić jeszcze jedną, ważną rzecz. Jedną z głównych reguł w systemach UNIX-owych jest to, aby nigdy nie uruchamiać programu z prawami roota jeśli nie jest to absolutnie konieczne. Tę regułę można zastosować również w naszym przypadku. Czemu? Przez większość czasu spędzanego przy komputerze jesteśmy zalogowani na konto zwykłego użytkownika, a jeśli chcemy posłuchać muzyki, należy mieć dostęp do odpowiednich urządzeń. Aby taki dostęp uzyskać, użytkownik musi być członkiem grupy "audio". Za chwilę dodamy użytkownika do tej grupy, dzięki czemu nie będzie miał problemów z korzystaniem z urządzeń dźwiękowych. Do zmiany przynależności użytkownika do grup używamy programu gpasswd. Do dokonania zmiany potrzebne są przywileje roota.

Listing 3.4: Dodawanie użytkownika do grupy audio

```
(Należy zastąpić słowo <username> odpowiednią nazwą użytkownika)
# gpasswd -a <username> audio
Adding user <username> to group audio
```

Sprawdzanie głośności

Teraz zakończyliśmy wszystkie przygotowania, możemy przystąpić do uruchomienia ALSA. Użytkownicy alsamixer mogą pominąć ten krok, program uruchomił dźwięk za nich.

Listing 3.5: Uruchamianie usług

```
(ALSA w modułach)
# /etc/init.d/alsasound start
 * Loading ALSA modules ...
 * Loading: snd-card-0 ... [ ok ]
 * Loading: snd-pcm-oss ... [ ok ]
 * Loading: snd-seq ... [ ok ]
 * Loading: snd-emu10k1-synth ... [ ok ]
 * Loading: snd-seq-midi ... [ ok ]
 * Restoring Mixer Levels ... [ ok ]
(Wbudowane sterowniki)
# /etc/init.d/alsasound start
 * Loading ALSA modules ...
 * Restoring Mixer Levels ... [ ok ]
```

Po zakończeniu etapu przygotowań możemy sprawdzić czy przypadkiem dźwięk nie jest całkowicie wyciszony. Użyjemy do tego programu alsamixer.

Listing 3.6: Uruchamianie alsamixer

```
(Otwiera konsolowy program, który pokazuje tylko wymagane ustawienia
# alsamixer
```

Ważne: Jeśli podczas pracy z alsamixer wystąpią błędy, takie jak np. "alsamixer: function snd_ctl_open

failed for default: No such file or directory", oznacza to najprawdopodobniej nieprawidłowo ustawione przez udev prawa dostępu do urządzeń. W celu przeladowania wpisów w /dev i naprawienia tego problemu należy wykonać polecenie killall udevd; udevstart.

Ostrzeżenie: Aby usłyszeć jakikolwiek dźwięk, należy włączyć oba kanały - PCM i Master i ustawić odpowiednio głośność.

- Do przemieszczania się pomiędzy kanałami służą klawisze strzałek w prawo i w lewo
- Aby zmienić ustawienie wyciszenia należy wybrać jakiś kanał, na przykład Master i wcisnąć klawisz m
- Do zmieniania poziomu głośności służą strzałki w górę i w dół

Uwaga: Należy uważać przy zmianie wartości Bass i Treble. 50 to w większości przypadków odpowiednie rozwiązanie. Zbyt wysokie ustawienia mogą skutkować irytującymi zniekształceniami dźwięku.

Po ustawieniu ALSA Mixer powinien wyglądać mniej więcej tak jak ten poniżej. Warto zwrócić uwagę na wartość 00 w miejscu MM ze zdjęcia wyżej oraz na dobrane odpowiednie ustawienia głośności.

Sprawdzanie dźwięku

Jeśli wszystko poszło dobrze, można wreszcie posłuchać muzyki. Nada się do tego zarówno jakiś mały i prosty program konsolowy, jak np. media-sound/madplay lub coś znacznie popularniejszego i bardziej skomplikowanego jak mpg123 czy xmms. Fani formatu ogg mogą użyć programu ogg123 z pakietu media-sound/vorbis-tools. Można użyć do tego dowolnego programu, do jego instalacji jak zwykle używamy polecenia emerge.

Listing 3.7: Instalowanie programów

```
(Instalowanie programu do odtwarzania plików .mp3)
# emerge madplay mpg123 xmms
(Instalowanie programu do odtwarzania plików .ogg)
# emerge vorbis-tools
```

Później odtwarzamy swoją ulubioną piosenkę:

Listing 3.8: Odtwarzanie muzyki

```
# madplay -v ~/mp3/Boys\ -\ Jesteś\ Szalona.mp3
MPEG Audio Decoder 0.15.2 (beta) - Copyright (C) 2000-2004 Robert Leslie et al.
Title: Jesteś Szalona
Artist: Boys
Album:
Year:
Genre: Disco Polo
00:04:19 Layer III, 160 kbps, 44100 Hz, joint stereo (MS), no CRC

# ogg123 ~/mp3/Boys\ -\ Jesteś\ Szalona.ogg
Audio Device: Advanced Linux Sound Architecture (ALSA) output

Playing: Boys - Jesteś Szalona.ogg
Ogg Vorbis stream: 2 channel, 44100 Hz
Genre: Disco Polo
Transcoded: mp3;160
Title: Jesteś Szalona
Artist: Boys
Date:
Album:
Time: 00:11.31 [04:28.75] of 04:40.06 (200.6 kbps) Output Buffer 96.9%
```

ALSA i flagi USE

Warto dodać flagę alsal do listy flag USE w pliku /etc/make.conf, aby upewnić się, że wszystkie programy mogące współpracować z ALSA będą zbudowane z jej obsługą. Niektóre architektury jak x86 czy amd64 mają tę flagę włączoną domyślnie.

Problemy?

Jeśli z jakichś powodów nie słyszać dźwięku, po pierwsze należy sprawdzić ustawienia [alsamixer](#). Większość problemów z dźwiękiem jest spowodowana jego wyciszeniem lub niedostatecznym zwiększeniem głośności. Warto również sprawdzić pod kątem tych samych niepożądanych ustawień aplikację zarządzającą poziomem głośności w używanym menedżerze okien.

System plików /proc również może okazać się tu przydatny, zwłaszcza jego podkatalog - /proc/asound. Pokróćce omówimy zawarte tam pozycje.

Listing 3.9: Informacje z /proc/asound

```
(Jeśli w pliku /proc/asound/cards znajduje się właściwa nazwa karty to
ALSA sobie z nią doskonale radzi)
# cat /proc/asound/cards
0 [Live] : EMU10K1 - Sound Blaster Live!
          Sound Blaster Live! (rev.6, serial:0x80271102) at 0xb800, irq 11

(Poniższe polecenie wyświetli wersję używanych sterowników ALSA w
przypadku, gdyby np. potrzebna była informacja na temat tego o ile są starsze w
porównaniu z tymi z pakietu alsa-driver)
# cat /proc/asound/version
Advanced Linux Sound Architecture Driver Version 1.0.8 (Thu Jan 13 09:39:32 2005 UTC).

(Szczegóły emulacji OSS przez ALSA)
# cat /proc/asound/oss/sndstat
Sound Driver:3.8.1a-980706 (ALSA v1.0.8 emulation code)
Kernel: Linux airwolf.zion 2.6.11ac1 #2 Wed May 4 00:35:08 IST 2005 i686
Config options: 0

Installed drivers:
Type 10: ALSA emulation

Card config:
Sound Blaster Live! (rev.6, serial:0x80271102) at 0xb800, irq 11

Audio devices:
0: EMU10K1 (DUPLEX)

Synth devices: NOT ENABLED IN CONFIG

Midi devices:
0: EMU10K1 MPU-401 (UART)

Timers:
7: system timer

Mixers:
0: SigmaTel STAC9721/23
```

Kolejnym bardzo częstym problemem jest pojawiająca się informacja o nieznanym symbolu w module - "Unknown symbol in module".

Listing 3.10: Nieznany symbol w module

```
# /etc/init.d/alsasound start
 * Loading ALSA modules ...
 *   Loading: snd-card-0 ...                                     [ ok ]
 *   Loading: snd-pcm-oss ...
WARNING: Error inserting snd_mixer_oss
(/lib/modules/2.6.12-gentoo-r6/kernel/sound/core/oss/snd-mixer-oss.ko): Unknown
symbol in module, or unknown parameter (see dmesg) FATAL: Error inserting
snd_pcm_oss
```

```
(/lib/modules/2.6.12-gentoo-r6/kernel/sound/core/oss/snd-pcm-oss.ko): Unknown
symbol in module, or unknown parameter (see dmesg) [ !! ]
* Loading: snd-mixer-oss ...
FATAL: Error inserting snd_mixer_oss
(/lib/modules/2.6.12-gentoo-r6/kernel/sound/core/oss/snd-mixer-oss.ko): Unknown
symbol in module, or unknown parameter (see dmesg) [ !! ]
* Loading: snd-seq ... [ ok ]
* Loading: snd-emu10k1-synth ... [ ok ]
* Loading: snd-seq-midi ... [ ok ]
* Restoring Mixer Levels ... [ ok ]
```

Jeśli przyjrzymy się uważnie dmesg, zauważymy takie wpisy:

Listing 3.11: Wyjście dmesg

(Wycięliśmy zbędne fragmenty wyniku poniższego polecenia)

```
# dmesg | less
ACPI: PCI Interrupt 0000:02:06.0[A] -> Link [APC3] -> GSI 18 (level, low) -> IRQ 209
snd_mixer_oss: Unknown symbol snd_unregister_oss_device
snd_mixer_oss: Unknown symbol snd_register_oss_device
snd_mixer_oss: Unknown symbol snd_mixer_oss_notify_callback
snd_mixer_oss: Unknown symbol snd_oss_info_register
snd_pcm_oss: Unknown symbol snd_unregister_oss_device
snd_pcm_oss: Unknown symbol snd_register_oss_device
snd_pcm_oss: Unknown symbol snd_mixer_oss_ioctl_card
snd_pcm_oss: Unknown symbol snd_oss_info_register
snd_mixer_oss: Unknown symbol snd_unregister_oss_device
snd_mixer_oss: Unknown symbol snd_register_oss_device
snd_mixer_oss: Unknown symbol snd_mixer_oss_notify_callback
snd_mixer_oss: Unknown symbol snd_oss_info_register
```

Taki problem pojawia się po przejściu ze sterowników z pakietu alsa-driver na te wbudowane w jądro, ponieważ mimo usunięcia alsa-driver, moduły wciąż pozostają w systemie dzięki ochronie plików konfiguracyjnych. Zatem mimo przejścia na moduły wbudowane w jądro polecenie modprobe wciąż będzie ładować mieszankę modułów z pakietu i z jądra.

Rozwiązanie jest proste, należy usunąć cały katalog z modułami tuż po odinstalowaniu pakietu alsa-driver. Warto się upewnić, że wersja jądra z poniższego wpisu jest prawidłowa.

Listing 3.12: Usuwanie modułów alsa-driver

```
# rm -rf /lib/modules/$(uname -r)/alsa-driver
```

Jednym z częstych powodów pojawiania się komunikatów błędu podobnych do tych podanych powyżej może być definiowanie parametru device_mode w plikach z katalogu /etc/modules.d gdy nie jest to konieczne. Należy przeszukać wszystkie pliki w tym katalogu i znaleźć ten, który jest winowajcą.

Listing 3.13: Poszukiwania device_mode

```
(Sprawdzamy w dmesg
# dmesg | grep device_mode
snd: Unknown parameter `device_mode'
(Teraz szukamy źródła błędu)
# grep device_mode /etc/modules.d/*
```

Zwykle wynikiem tego poszukiwania będzie plik o nazwie alsa wewnętrz którego znajdować będzie się linia options snd device_mode=0666. Należy ją usunąć i ponownie uruchomić skrypt startowy alsasound.

4. Dodatkowe funkcje ALSA

Obsługa MIDI

Jeśli karta dźwiękowa posiada syntezator MIDI i jeśli chcemy korzystać z dźwięków zapisanych w formacie .mid, będziemy musieli wykorzystać zestaw programów o nazwie awesfx kontrolujący sterowniki AWE32. Po pierwsze musimy go zainstalować. Jeśli karta nie posiada sprzętowego syntezatora można użyć wirtualnego.Więcej informacji o nim znajduje się w akapicie wirtualne syntezatory.

Listing 4.1: Instalowanie awesfx

```
# emerge awesfx
```

Uwaga: Potrzebne będzie też skopiowanie plików SoundFont (SF2) z płyty ze sterownikami do karty dźwiękowej do katalogu /usr/share/sounds/sf2/. Przykładowa nazwa pliku SoundFont dla karty Creative SBLive! to 8MBGMSFX.SF2.

Po przekopiowaniu plików SoundFont można odtwarzać pliki tak jak w poniższym przykładzie. Warto dodać polecenie asfxload do skryptu /etc/conf.d/local.start tak, aby pliki te były wczytywane przy każdym uruchomieniu komputera.

Uwaga: Ścieżki do katalogów w /mnt niemal na pewno będą inne, podajemy tu tylko przykład, wszystkie polecenia należy odpowiednio skorygować.

Listing 4.2: Przegrywanie plików soundfont

```
(Najpierw kopujemy plik)
# cp /mnt/win2k/Program\ Files/CreativeSBLive2k/SFBank/8MBGMSFX.SF2 /usr/share/sounds/sf2/
(Jest też na płycie SoundBlastera)
# cp /mnt/cdrom/AUDIO/ENGLISH/SFBANK/8MBGMSFX.SF2 /usr/share/sounds/sf2/
(A następnie wczytujemy)
# asfxload /usr/share/sounds/sf2/8MBGMSFX.SF2
```

Od tej pory można odtwarzać midi za pomocą programów takich jak np. aplaymidi. Polecenie aplaymidi -l wyświetli listę dostępnych portów, należy wybrać jeden do odtworzenia określonego pliku.

Listing 4.3: Odtwarzanie MIDI

```
(Sprawdzanie otwartych portów)
# aplaymidi -l
Port      Client name                                Port name
64:0      EMU10K1 MPU-401 (UART)                      EMU10K1 MPU-401 (UART)
65:0      Emu10k1 WaveTable                           Emu10k1 Port 0
65:1      Emu10k1 WaveTable                           Emu10k1 Port 1
65:2      Emu10k1 WaveTable                           Emu10k1 Port 2
65:3      Emu10k1 WaveTable                           Emu10k1 Port 3
(Wybieramy port i odtwarzamy plik)
# aplaymidi --port=65:0 ~/mid/jestes_szalona.mid
```

Wirtualne syntezatory

Jeśli karta nie posiada sprzętowego syntezatora potrzebny będzie wirtualny, na przykład timidity++. Jego instalacja to prosta sprawa.

Listing 4.4: Instalowanie timidity++

```
# emerge timidity++
```

Aby timidity odgrywało dźwięki również potrzebny jest soundfont. Jeśli nie dysponuje się akurat żadnym, należy zainstalować pakiet timidity-eawpatches lub timidity-shompatches, zawierający odpowiednie pliki. Można posiadać kilka wersji soundfontów, a własną konfigurację umieszcza się w pliku /usr/share/timidity/. Zmiany pomiędzy jednym, a drugim ustawieniem dla timidity dokonuje się za pomocą polecenia timidity-update, które jest dostępne po zemergowaniu pakietu timidity++.

Listing 4.5: Instalowanie soundfontów

```
# emerge timidity-eawpatches
# timidity-update -g -s eawpatches

(lub)

# emerge timidity-shompatches
# timidity-update -g -s shompatches
```

Warto dodać skrypt timidity do domyślnego poziomu uruchomieniowego.

Listing 4.6: Dodawanie timidity do domyślnego poziomu uruchomieniowego

```
# rc-update add timidity default
# /etc/init.d/timidity start
```

Od tej pory można już odtwarzać pliki MIDI.

Dodatkowe narzędzia i firmware

Niektóre karty mogą skorzystać dzięki instalacji narzędzi z pakietów als-tools i als-firmware. Przed zainstalowaniem pakietu als-tools należy się upewnić, że zmienna `ALSA_TOOLS` w pliku `/etc/make.conf` zawiera nazwy wszystkich niezbędnych narzędzi. Na przykład:

Listing 4.7: Wybieranie narzędzi dla ALSA w /etc/make.conf

```
ALSA_TOOLS="as10k1 ac3dec"
```

Jeśli zmienna `ALSA_TOOLS` nie zostanie ustawiona to zostaną zbudowane wszystkie programy. Ostatnią czynnością opisaną w tym tekście jest zainstalowanie pakietów als-tools (i als-firmware - jeśli również jest konieczny).

Listing 4.8: Instalowanie narzędzi ALSA

```
# emerge als-tools
```

12. Java w Gentoo

1. Co to jest Java?

Ogólny przegląd

Java to język programowania opracowany przez inżynierów z laboratoriów Sun Microsystems. Jest to zorientowany obiektowo język, który może być uruchamiany na wielu platformach bez konieczności rekompilacji kodu dla każdej z platform osobno. Pomimo, że Java może być kompilowana do zwykłego programu, to jej największą popularność powoduje fakt, że jest bardzo przenośna oraz posiada możliwość łatwego pozbywania się śmieci z kodu. Aby osiągnąć taki stopień przenośności kompilatory Java kompilują kod do postaci "Java bytecode", który można uruchomić w JRE (Java Runtime Environment - środowisko uruchomieniowe Java) a nie bezpośrednio w systemie operacyjnym.

Aby uruchomić program napisany w Java, należy mieć zainstalowany JRE (Java Runtime Environment - środowisko uruchomieniowe Java). JRE instaluje główne biblioteki, maszynę wirtualną Java, wtyczki do przeglądarki, etc. JDK (Java Development Kit) dodaje narzędzia programistyczne, jak na przykład kompilator kodu Java czy debugger.

2. Wprowadzenie

Istniejące instalacje

Dla istniejących instalacji, bez względu na to czy mieliśmy zainstalowane jakieś kolwiek pakiety związane z Java, powinniśmy się upewnić, że postępowaliśmy zgodnie z dokumentem [Aktualizacja Javy](#).

Nowe instalacje

W przypadku nowej instalacji nie musimy wykonywać żadnych dodatkowych czynności.

3. Instalacja wirtualnej maszyny

Wybory

W Gentoo dostępne jest kilka rodzajów zarówno JRE (Runtime Environment) jak i JDK (Development Kit).

Twórcy	JDK	JRE
The Blackdown Java Kit	dev-java/blackdown-jdk	dev-java/blackdown-jre
Sun's Java Kit	dev-java/sun-jdk	dev-java/sun-jre-bin
The IBM Java Kit	dev-java/ibm-jdk-bin	dev-java/ibm-jre-bin
The Compaq Java Kit dla Alpha/Linux/GNU	dev-java/compaq-jdk	dev-java/compaq-jre
BEA WebLogic's J2SE Development Kit	dev-java/jrockit-jdk-bin	

Domyślnym wyborem jest wersji 1.4 Blackdown JRE/JDK, ponieważ jest ona darmowa i dostępna bez przeprowadzania zbędnej rejestracji.

Zarówno Sun jak i IBM są zdecydowanie szybsze, niemniej jednak, zdobycie ich wymaga nieco więcej pracy, gdyż należy przeczytać i zaakceptować ich licencję przed pobraniem plików z sieci (IBM dodatkowo wymaga rejestracji).

Instalowanie JRE/JDK

Aby zainstalować domyślne dla wykorzystywanego profilu JDK, należy uruchomić emerge virtual/jdk, natomiast by zainstalować domyślne JRE - emerge virtual/jre.

Aktualnie Sun zmienił licencję na swoje pakiety JDK i JRE na bardziej przyjazne dla dystrybucji Linuksa. W wyniku tego od wersji 1.5, mogą one być pobierane za darmo, bez niepotrzebnego zwracania głowy.

Uwaga: JDK zawiera w sobie JRE, więc jeżeli zainstalujemy JDK, nie powinniśmy dodatkowo instalować JRE.

Instalacja maszyn wirtualnych pobieranych ręcznie

Jak wcześniej wspomnialiśmy, niektóre z pakietów JDK lub JRE wymagają podjęcia kilku kroków przed ich instalacją. Uruchamiamy emerge tak jakbyśmy normalnie instalowali te pakiety. Po wydaniu tego polecenia zostaniemy poinformowani co i skąd mamy pobrać.

Wszystkie wskazane pliki należy umieścić w katalogu /usr/portage/distfiles. Dopiero potem można ponownie uruchomić polecenie

emerge, aby dane JRE/JDK zostały poprawnie zainstalowane.

4. Konfiguracja wirtualnej maszyny

Przegląd

Gentoo posiada możliwość bezkonfliktowego zainstalowania wielu JDK oraz JRE równocześnie.

Używając z konta roota narzędzia java-config, można ustawić domyślne środowisko dla całego systemu. Użytkownicy mogą użyć java-config, aby ustawić własne domyślne środowisko, które może być inne niż to określone w systemie.

Uwaga: Środowisko dla systemu oraz maszynę wirtualną możemy zmienić również używając narzędzia eselect. Polecenie eselect java-vm help wyświetli więcej informacji na ten temat.

Ustawianie domyślnego środowiska wirtualnej maszyny

Uruchamiając polecenie java-config --list-available-vms, wyświetlona zostanie lista dostępnych JRE oraz JDK na danym systemie. Poniżej przedstawiamy przykład takiej listy:

Listing 4.1: Wyświetlanie dostępnych maszyn wirtualnych

```
# java-config --list-available-vms
1) Blackdown JDK 1.4.2.03 [blackdown-jdk-1.4.2]
2) Blackdown JRE 1.4.2.03 [blackdown-jre-1.4.2]
3) Blackdown 32bit JRE 1.4.2.03 [emul-linux-x86-java-1.4.2]
4) Sun 32bit JRE 1.5.0.08 [emul-linux-x86-java-1.5]
5) Kaffe 1.1.7 [kaffe]
*) Sun JDK 1.5.0.08 [sun-jdk-1.5]
```

Znak * (gwiazdka) informuje nas o aktualnie aktywnej maszynie wirtualnej (systemowej lub wybranej przez użytkownika, jeżeli taka jest). Nazwy w nawiasach ("[]") to wskaźnik lub identyfikator dla konkretnej maszyny wirtualnej. Przekazując tę wartość (lub numer) do java-config --set-system-vm ustawia się wskazaną maszynę wirtualną jako domyślną. Poniżej znajduje się przykład takiej czynności:

Listing 4.2: Ustawianie domyślnej maszyny wirtualnej dla systemu

```
(Poprzez wartość z nawiasów (zalecane))
# java-config --set-system-vm blackdown-jdk-1.4
Now using blackdown-jdk-1.4 as your generation-2 system JVM
(Poprzez numer maszyny)
# java-config --set-system-vm 2
Now using sun-jdk-1.5 as your generation-2 system JVM
```

Jako zwykły użytkownik możemy użyć java-config --set-user-vm

Uwaga: Nie musimy wykonywać source profilu w celu aktualizacji maszyny wirtualnej systemowej lub użytkownika.

Preferowana maszyna wirtualna

Podczas instalowania pakietów Javy, wirtualna maszyna może być - i będzie zmieniona - w razie potrzeby.

Ponieważ istnieje dużo różnych wirtualnych maszyn Javy, nie jesteśmy w stanie przetestować działania każdego pakietu na każdej z nich. Aby się upewnić, że każdy z nich został zainstalowany poprawnie, wybraliśmy listę domyślnych/wspieranych maszyn wirtualnych dla każdej architektury. Możemy znaleźć je w /usr/share/java-config/config-jdk-defaults.conf. Kiedy instalujemy pakiety w Javie, a w systemie zostanie wykryta chociaż jedna maszyna zgodna z tym plikiem, zostanie ona automatycznie użyta zamiast maszyny systemowej.

Zmiana maszyny wirtualnej podczas instalacji jest potrzebna również w przypadku, gdy maszyna systemowa ma wersję 1.4 natomiast instalowany pakiet wymaga wersji 1.5. Podczas instalacji zostanie ona zmieniona na 1.5, pozostawiając maszynę systemową nietkniętą.

Oczywiście, w Gentoo wszystko zależy od nas, więc możemy ominąć te domyślne ustawienia w /etc/java-config-2/build/jdk.conf i mieć pełną kontrolę nad tym jaką maszyną wirtualną jest używana w naszym systemie. Kilka przykładów:

Listing 4.3: Przykład /etc/java-config-2/build/jdk.conf

```
(Zawsze będzie korzystać z sun-jdk, np. sun-jdk-1.4 for 1.4, sun-jdk-1.5 for 1.5, etc)
*=sun-jdk
```

Listing 4.4: Przykład /etc/java-config-2/build/jdk.conf

```
(Będzie korzystać sun-jdk-1.5 kiedy tylko jest to możliwe, poza przypadkami gdzie użycie
1.3 lub 1.4 jest konieczne)
*=sun-jdk-1.5
```

Listing 4.5: Przykład /etc/java-config-2/build/jdk.conf

```
# Dla 1.3 preferujemy sun-jdk 1.4, a gdy niedostępne ibm-jdk-bin,
# Dla 1.4 użyjemy blackdown-jdk, a dla 1.5 sun-jdk
1.3=sun-jdk-1.4 ibm-jdk-bin
1.4=blackdown-jdk
1.5=sun-jdk
```

Ostrzeżenie: Nie musimy edytować tego pliku. Jeżeli chcemy zmienić te opcje w celu użycia nie wspieranej maszyny wirtualnej, prawdopodobnie zakończy się to błędem. Zgłoszone błędy z użyciem niewspieranych maszyn wirtualnych będą miały niższy priorytet niż te ze wspieranymi.

5. Kompilatory

Standardowym kompilatorem Java używanym do budowania jest javac, który wchodzi w skład każdego JDK. Podobnie jak wybieraliśmy maszynę wirtualną, możemy również wybrać, którego kompilatora chcemy użyć. W pliku /etc/java-config-2/build/compilers.conf zdefiniujemy listę ustawień, w których zdecydujemy jakiego kompilatora chcemy użyć.

Listing 5.1: /etc/java-config-2/build/compilers.conf

```
# If the ebuild supports it
# it will check the COMPILERS var front to back and
# use the first compiler that is installed

COMPILERS="ecj-3.1 jikes javac"
```

Niektóre kompilatory nie wspierają wszystkich argumentów -target i -source. Dlatego każdy kompilator z powyższej listy jest sprawdzany pod kątem wspierania danego -source/-target. javac będzie działać we wszystkich przypadkach, więc jeżeli żaden kompilator z listy nie będzie spełniał wymagań, zostanie użyty javac.

Więcej informacji n/t kompilatorów znajdziemy poniżej:

Nazwa	Identyfikator	Pakiet	Opis
javac	javac	N/A	Jest to domyślny kompilator, wchodzący w skład wszystkich JDK.
jikes	jikes	dev-java/jikes	Jikes był początkowo tworzony przez IBM. Najśmieszniejsze jest to, że jest on ogólnie szybszy od javac. Należy jednak zauważać, że jest on bardziej pedantyczny i będzie wyświetlał błędy w przypadkach, w których javac nie ma problemów. Dodatkowo, jikes nie wspiera Javy w wersji 1.5
Eclipse Compiler dla Java	ecj-3.1	=dev-java/eclipse-ecj-3.1*	ECJ jest kompilatorem używanym przez środowisko Eclipse. Zawiera on bardzo dużo udoskonalień i jest w miarę szybki. Wspiera składnię Java 1.5

6. Ustawianie domyślnej CLASSPATH

Ostrzeżenie: Opcje objaśnione w tym akapicie, nie powinny zostać stosowane pochopnie i być może zostaną usunięte w przyszłości. Nie polecamy ich używania, chociażby ze względu na to, że nasze projekty lub aplikacje w Java mogą doskonale zarządzać swoimi zmiennymi CLASSPATH. Jeżeli zdecydujemy się na ustawienie domyślnej wartości CLASSPATH, niektóre aplikacje mogą zachowywać się w nieprzewidziany sposób, ponieważ natrafią na znajdujące się w CLASSPATH nieznane klasy.

java-config możemy użyć zarówno do ustawienia systemowej CLASSPATH, jak i oddzielnej dla każdego użytkownika

Po pierwsze powinniśmy wylistować dostępne biblioteki Java zainstalowane w naszym systemie, które możemy włożyć do CLASSPATH. Poniżej znajduje się przykładowe wyjście programu:

Listing 6.1: Lista klas

```
# java-config --list-available-packages
[xerces-2] The next generation of high performance, fully compliant XML parsers in the Apache Xerces project
[junit] Simple framework to write repeatable tests (/usr/share/junit/package.env)
[bsh] BeanShell: A small embeddable Java source interpreter (/usr/share/bsh/package.env)
[bcel] The Byte Code Engineering Library: analyze, create, manipulate Java class files (/usr/share/bcel/package.env)
[log4j] A low-overhead robust logging package for Java (/usr/share/log4j/package.env)
...
...
```

Znow, nazwy w nawiasach ([]) są identyfikatorem, który musimy wpisać w java-config --set-system-classpath tak jak pokazano poniżej:

Listing 6.2: Ustawianie ścieżek dla klas

```
# java-config --set-system-classpath log4j,xerces-2
```

Uwaga: Obecny katalog(.) nie będzie należał do systemowego classpath, katalog ten powinien zostać dodany przez systemowy profil logowania.

Teraz powinniśmy zaktualizować nasze środowisko, poprzez przelogowanie lub wykonanie polecenia source /etc/profile.

Jeśli chodzi o użytkowników, java-config --set-user-classpath stworzy ~/.gentoo/java-env-classpath, który powinien być teraz wczytywany przez plik konfiguracyjny powłoki.

Listing 6.3: Wczytywanie własnych classpath

```
if [[ -f "${HOME}/.gentoo/java-env-classpath" ]]; then
    source ${HOME}/.gentoo/java-env-classpath
fi
```

Jeżeli bardzo chcemy zmienić systemową, lub domyślną dla użytkownika classpath, możemy dodać coś takiego do pliku konfiguracyjnego powłoki (jednak dalej nie polecamy tego rozwiązania):

Listing 6.4: Ustawianie classpath

```
# export CLASSPATH="${CLASSPATH}:$(java-config --classpath log4j,xerces-2)"
```

7. Wtyczki Java dla przeglądarek

Instalacja wtyczki

Wtyczkę Java dla naszej przeglądarki możemy zainstalować instalując wirtualną maszynę z flagą USE nsplugin.

Uwaga: Flaga nsplugin nie jest dostępna dla wszystkich architektur. Dostępność wtyczki sprawdzamy przed instalacją przez wydanie polecenia emerge -pv <java-vm>.

Portage pozwala na instalację kilku wersji wtyczki Java, jednak używana będzie tylko jedna, przeznaczona dla naszej przeglądarki. Listę dostępnych wtyczek możemy przejrzeć po wydaniu komendy:

Listing 7.1: Wyświetlanie listy dostępnych wtyczek

```
# eselect java-nsplugin list
[1] sun-jre-bin-1.5
[2] blackdown-jre-1.4.2
```

W tym przykładzie, za wtyczkę dla przeglądarki wybieramy sun-jre-bin.

Listing 7.2: Wybieranie wtyczki

```
# eselect java-nsplugin set sun-jre-bin-1.5
Musimy zweryfikować czy wybrana została prawidłowa wtyczka.
```

Listing 7.3: Weryfikacja wtyczki

```
# eselect java-nsplugin list
[1] sun-jre-bin-1.5 current
[2] blackdown-jre-1.4.2
```

W stronie Java.com możemy znaleźć również stronę [weryfikującą zainstalowaną wtyczkę](#). Dodatkowo, gdy używamy przeglądarki bazującej na rozwiązańach Mozilla, weryfikację możemy przeprowadzić poprzez wpisanie about:plugins w pasku adresu.

Wtyczki na komputerach z multilib

Jeśli używamy systemu zarówno z 64 jak i 32-bitowymi bibliotekami (dla przykładu, na architekturze AMD64), możemy mieć zainstalowaną zarówno wtyczkę 64 jak i 32-bitową.

Sprawdzamy dostępne w systemie wtyczki:

Listing 7.4: Wyświetlanie listy dostępnych wtyczek

```
# eselect java-nsplugin list
Available 32-bit Java browser plugins
[1] emul-linux-x86-java-1.4.2
[2] emul-linux-x86-java-1.5
Available 64-bit Java browser plugins
[1] blackdown-jdk-1.4.2
[2] blackdown-jre-1.4.2
```

Dla przeglądarki 32-bitowej (dla przykładu firefox-bin), możemy wybrać 32-bitową wersję wtyczki, natomiast dla przeglądarki 64-bitowej (na przykład konqueror) wybieramy 64-bitową wersję wtyczki.

Listing 7.5: Wybieranie wtyczek

```
# eselect java-nsplugin set 32bit emul-linux-x86-java-1.4.2
# eselect java-nsplugin set 64bit blackdown-jdk-1.4.2
```

Powinniśmy zweryfikować czy zostały wybrane odpowiednie wtyczki:

Listing 7.6: Weryfikacja wtyczek

```
# eselect java-nsplugin list
Available 32-bit Java browser plugins
[1] emul-linux-x86-java-1.4.2 current
[2] emul-linux-x86-java-1.5
Available 64-bit Java browser plugins
[1] blackdown-jdk-1.4.2 current
[2] blackdown-jre-1.4.2
```

8. Flagi USE których można użyć z Javą

Ustawianie flag USE

Aby uzyskać więcej informacji dotyczących flag USE, należy odwołać się do rozdziału opisującego [flagi USE](#) w Podręczniku Gentoo.

Flagi

- Flaga **java** dodaje obsługę dla Javy w wielu programach
- Flaga **nsplugin** dodaje obsługę java dla przeglądarek z rodziny Mozilli (włączając w to Firefox). Flaga ta jest niezbędna gdy zechcemy przeglądać aplikacje java w przeglądarce z tej rodziny.
- Flaga **source** zainstaluje spakowane wersje kodów źródłowych pakietów. Jest to najczęściej używane w IDE do dołączenia kodów źródłowych do bibliotek, których używamy.
- Dla pakietów Java, flaga **doc** instaluje po prostu dokumentację dla API używając javadoc.

13. Instalacja i konfiguracja CUPS

1. Gentoo i drukarki

Wybór właściwych narzędzi

Linux doskonale radzi sobie z drukarkami dzięki narzędziu o nazwie CUPS ([Common Unix Printing System](#)), zajmującemu się ich obsługą. Od czasu powstania w 1999 projekt ten znacznie się rozwinął i teraz pozwala na znacznie łatwiejszą instalację i konfigurację niż kiedyś.

W tym tekście omówimy konfigurację drukarek sieciowych i lokalnych wyłącznie za pomocą systemu CUPS. W związku z tym, że projekt ten posiada doskonałą dokumentację nie będziemy zbytnio wdawać się w techniczne szczegóły jego pracy, ani omawiać bardzo zaawansowanych rozwiązań.

2. Konfiguracja jądra

Wprowadzenie

Pierwsza informacja jaką jest potrzebna to sposób podłączenia drukarki do komputera. Są trzy możliwości: drukarka podłączona poprzez port LPT, przez port USB lub drukarka sieciowa. Jeśli jest to drukarka sieciowa ważna jest także informacja o tym z jakiego protokołu ona korzysta, czyli czy używa IPP (Internet Printing Protocol) czy Microsoft Windows SMB-CIFS (MS Windows Sharing).

W kilku kolejnych akapitach omówimy konfigurację jądra jaką jest wymagana dla nowo tworzonego systemu drukowania. Sposób konfiguracji jest zależny od typu połączenia drukarki z komputerem, instrukcje dla każdego z nich znajdują się w osobnym akapicie:

- Drukarka lokalna (LPT)
- Drukarka lokalna (USB)
- Drukarka sieciowa (IPP)
- Drukarka sieciowa (SMB-CIFS)

Przechodzimy do katalogu `/usr/src/linux` i wpisujemy polecenie `make menuconfig`, aby rozpocząć konfigurowanie jądra. Jeśli do budowy kernela był używany program `genkernel` również należy wykonać kilka kolejnych czynności tak, aby upewnić się, że wybrane ustawienia są prawidłowe.

W kolejnych przykładowych konfiguracjach wbudujemy obsługę wymaganych sterowników na stałe, rezygnując z tworzenia ich w postaci modułów. Nie jest to obowiązkowe, można również z łatwością korzystać ze sterowników w modułach, należy się jednak wtedy upewnić, że wszystkie konieczne sterowniki są ładowane. W przykładach korzystamy również tylko i wyłącznie z jądra serii 2.6. Użytkownicy 2.4 mogą uznać za konieczne poszukanie innego opisu konfiguracji.

Pora przejść do odpowiedniego akapitu i skonfigurować (lub tylko sprawdzić) jądro.

Drukarka sieciowa (LPT)

Port LPT jest używany do identyfikacji równoległego portu drukarki. Po pierwsze trzeba zatem włączyć obsługę portu równoległego, a następnie wybrać opcję dla portu w stylu PC (no chyba, że jest się użytkownikiem SPARC).

Listing 2.1: Konfiguracja portu równoległego

```
Device Drivers -->
  <*> Parallel port support
  <*>   PC-style hardware

Device Drivers -->
  Character Devices -->
    <*> Parallel printer support
    [*]     IEEE 1284 transfer modes)
```

Uwaga: Część użytkowników może również potrzebować innych opcji z zakładki Parallel port support. Więcej informacji o poszczególnych opcjach znajduje się w pomocy dostępnej po wybraniu opcji Help.

I to wszystko. Należy zakończyć konfigurację jądra i je przebudować. Warto pamiętać o przekopiowaniu obrazu jądra do katalogu `/boot` (i nie zapomnieć o podmontowaniu przedtem partycji `/boot`, jeśli to konieczne), a następnie uaktualnić konfigurację programu ładującego. Na koniec należy ponownie uruchomić komputer i uruchomić Gentoo przy pomocy nowego jądra.

Kolejny etap to instalowanie i konfigurowanie CUPS.

Drukarka lokalna (USB)

Do uruchomienia drukowania przez port USB potrzebna jest obsługa dwóch rzeczy: odpowiedniej magistrali USB oraz drukarek USB.

Listing 2.2: Konfiguracja portu USB dla drukarki

```
Device Drivers -->
  USB Support -->
    <*> Support for Host-side USB
    (...)

    --- USB Host Controller Drivers
    (Należy wybrać HCD, z którego korzysta komputer, z pomocą w
     wybraniu właściwego może przyjść polecenie "lspci -v | grep HCI")
    <*> EHCI HCD (USB 2.0) support ( lub )
    <*> OHCI HCD support ( lub )
    <*> UHCI HCD (most Intel and VIA) support
    (...)

    <*> USB Printer support
```

To wszystko. Należy zakończyć konfigurację jądra i je przebudować. Warto pamiętać o przekopiowaniu obrazu jądra do katalogu /boot (i nie zapomnieć o podmontowaniu przedtem partycji /boot, jeśli to konieczne), a następnie uaktualnić konfigurację programu ładującego. Na koniec należy ponownie uruchomić komputer i uruchomić Gentoo przy pomocy nowego jądra.

Kolejny etap to instalowanie i konfigurowanie CUPS.

Drukarka sieciowa (IPP)

W celu połączenia się z drukarką sieciową poprzez IPP potrzebna jest prawidłowa konfiguracja urządzeń sieciowych. Zakładając, że jądro już je posiada możemy spokojnie przejść do akapitu dotyczącego [instalowania i konfigurowania CUPS](#).

Drukarka sieciowa (SMB-CIFS)

Jądro musi posiadać obsługę SMB CIFS:

Listing 2.3: Konfiguracja drukarki SMB-CIFS

```
File systems -->
  Network File Systems -->
    <*> SMB file system support (to mount Windows shares etc.)
    <*> CIFS support (advanced network file system for Samba, Windows and other CIFS compliant
```

To wszystko. Należy zakończyć konfigurację jądra i je przebudować. Warto pamiętać o przekopiowaniu obrazu jądra do katalogu /boot (i nie zapomnieć o podmontowaniu przedtem partycji /boot, jeśli to konieczne), a następnie uaktualnić konfigurację programu ładującego. Na koniec należy ponownie uruchomić komputer i uruchomić Gentoo przy pomocy nowego jądra.

Kolejny etap to instalowanie i konfigurowanie CUPS.

3. Instalowanie i konfigurowanie CUPS

Instalacja

Instalowanie CUPS w Gentoo jest dość proste, a CUPS posiada kilka opcji, które na pewno zainteresują sporą część użytkowników. Żeby je włączyć należy skorzystać z flag USE z nimi powiązanych.

Flaga USE	Wpływ na CUPS
nls	Włącza obsługę języków narodowych. Dzięki tej flagie CUPS jest w stanie zapewnić zlokalizowany interfejs pozwalający na obsługę systemu w języku ojczystym.
pam	Pozwala na uwierzytelnianie zadań dla drukarki poprzez moduły PAM.
samba	Do współdzielenia drukarki z komputerami pracującymi w sieci pod kontrolą Windows za pomocą protokołu SMB-CIFS
slp	W zarządzanym środowisku może zaistnieć potrzeba udostępnienia usługi innym podmiotom. Dzięki protokołowi SLP można w łatwy sposób odnaleźć i skonfigurować taką usługę w swoim systemie. Jeśli drukarka jest dostępna jako usługa lub też będzie w ten sposób udostępniana należy włączyć tę flagę USE.
ssl	Jeśli chcemy zapewnić bezpieczeństwo zdalnego uwierzytelniania i drukowania musimy na wszystkich komputerach w sieci posiadać obsługę SSL, pozwalającą na tworzenie szyfrowanych sesji wydruku.

Należy sprawdzić obecne ustawienia USE. Zmiany ich ustawień dokonujemy w pliku /etc/portage/package.use, tak by dotyczyły tylko CUPS, a nie całego systemu.

Listing 3.1: Ustawienia flag USE dla CUPS

```
# emerge -pv cups
[ebuild N      ] net-print/cups-1.1.23-r1 +nls +pam -samba -slp +ssl 0 kB

(Na przykład by włączyć obsługę SAMBA)
# vim /etc/portage/package.use
net-print/cups samba
```

Kiedy wszystko będzie gotowe można przystąpić do instalacji CUPS.

Listing 3.2: Instalowanie CUPS

```
# emerge cups
```

Po instalacji Portage poprosi o wykonanie polecenia rc-update add coldplug default co umożliwia drukarkom USB korzystanie z niezbędnych modułów jądra.

Jeśli jest to drukarka lokalna to CUPS musi być uruchamiany automatycznie przy starcie systemu. Należy się upewnić, że jest włączona zanim zacznie się uruchamianie CUPS.

Listing 3.3: Automatyczne uruchamianie CUPS

```
# /etc/init.d/cupsd start
# rc-update add cupsd default
```

Konfiguracja

Domyślna konfiguracja serwera CUPS znajduje się w pliku /etc/cups/cupsd.conf i jest odpowiednia dla większości użytkowników. Część z nich może jednak chcieć tę konfigurację dostosować do własnych potrzeb.

W kolejnym akapicie omówimy kilka najczęściej wprowadzanych zmian.

- Zdalny dostęp do drukarki: Pozwalamy innym użytkownikom na korzystanie z naszej drukarki.
- Zdalna administracja CUPS: Uzyskujemy dostęp do narzędzi administracyjnych CUPS z innych komputerów.
- Obsługa sterowników Windows PCL: Konfigurujemy CUPS do pracy ze sterownikami Windows PCL, co pozwala na współdzielenie drukarki poprzez Sambę.
- Konfiguracja drukarki sieciowej: Konfigurujemy system tak, aby używał drukarki podłączonej do innego komputera, nie pracującego pod kontrolą Windows.

Zdalny dostęp do drukarki

Jeśli chcemy mieć dostęp do drukarki z innych komputerów poprzez protokół IPP to musimy nadać im do niej prawa dostępu, czego dokonuje się w pliku /etc/cups/cupsd.conf. Jeśli drukarka ma być współdzielona poprzez Sambę to zmiana ta nie jest konieczna.

Ulubionym edytorem otwieramy plik /etc/cups/cupsd.conf i dodajemy tam linię Allow dla każdego komputera, który ma posiadać możliwość łączenia się z drukarką. W poniższym przykładzie damy dostęp do drukarki komputerom z sieci lokalnej, czyli takim, których adresy IP rozpoczynają się od ciągu znaków 192.168.0.

Listing 3.4: Umożliwianie zdalnego dostępu do drukarki

```
<Location />
Order Deny,Allow
Deny From All
Allow From 127.0.0.1
Allow From 192.168.0./*
</Location>
```

Jeśli CUPS ma odpowiadać na żądania drukowania z innych komputerów w sieci, należy ustawić mu port, na którym takich żądań ma

oczekiwać.

Listing 3.5: Konfigurowanie portu w /etc/cups/cupsd.conf

```
Port 631
(Poniższe linie muszą być zakomentowane)
#Listen 127.0.0.1:631
#Listen localhost:631
```

Zdalne administrowanie CUPS

Użytkownicy zainteresowani zdalnym administrowaniem systemem drukowania muszą zapewnić dostęp do interfejsu administratora z komputerów innych niż localhost. W tym celu należy edytować plik /etc/cups/cupsd.conf i wyszczególnić wszystkie adresy, z których zamierzamy dokonywać zmian administracyjnych. Na przykład, aby zapewnić taki dostęp komputerowi o adresie IP 192.168.0.3 musimy dokonać następującego wpisu:

Listing 3.6: Zapewnianie zdalnego dostępu w /etc/cups/cupsd.conf

```
<Location /admin>
(...)
Order Deny,Allow
Deny From All
Allow From 127.0.0.1
Allow From 192.168.0.3
</Location>
```

Po dokonaniu zmian w pliku /etc/cups/cupsd.conf należy ponownie uruchomić CUPS za pomocą polecenia /etc/init.d/cupsd restart.

Obsługa sterowników Windows PCL

Sterowniki PCL wysyłają do serwera druku dane w postaci pierwotnej. Aby umożliwić drukowanie takich danych w CUPS należy wyedytować pliki /etc/cups/mime.types oraz /etc/cups/mime.convs i odkomentować w nich linię application/octet-stream.

Listing 3.7: Włączanie obsługi drukowania pierwotnego

```
# vim /etc/cups/mime.types
(Odkomentowujemy poniższą linię)
application/octet-stream

# vim /etc/cups/mime.convs
(Również odkomentowujemy linię z poniższym wpisem)
application/octet-stream      application/vnd.cups-raw      0      -
```

Po dokonaniu zmian w pliku /etc/cups/cupsd.conf należy ponownie uruchomić CUPS za pomocą polecenia /etc/init.d/cupsd restart.

Konfigurowanie drukarki sieciowej

Jeśli system ma korzystać z drukarki sieciowej wystarczy dopisać mu odpowiednie urządzenie w pliku /etc/cups/client.conf.

Zakładając, że drukarka jest podłączona do komputera o nazwie printserver.mydomain, należy dokonać następującego wpisu do pliku /etc/cups/client.conf:

Listing 3.8: Edytowanie pliku client.conf

```
# vim /etc/cups/client.conf

(Oczywiście wpis printserver.mydomain należy zastąpić nazwą
odpowiedniego serwera wydruku)
ServerName printserver.mydomain
```

Serwer wydruku będzie przekierowywał nasze zadania do swojej domyślnej drukarki. Aby zmienić domyślną drukarkę należy

skorzystać z polecenia lpoptions:

Listing 3.9: Zmiana domyślnej drukarki

```
(Lista dostępnych urządzeń)
# lpstat -a
hpljet5p accepting requests since Jan 01 00:00
hpjet510 accepting requests since Jan 01 00:00

(Ustawienie HP LaserJet 5P jako drukarki domyślnej)
# lpoptions -d hpljet5p
```

4. Konfigurowanie drukarki

Wprowadzenie

Jeśli drukarka, którą chcemy skonfigurować jest dostępna zdalnie, przez serwer działający pod kontrolą CUPS to wystarczy zastosować się do wskazówek znajdujących się w akapicie zatytułowanym [Konfigurowanie drukarki sieciowej](#).

Wykrywanie drukarki

Jeśli drukarka, obojętnie czy podłączona przez USB, czy przez port równoległy, jest włączona podczas uruchamiania Gentoo będzie można uzyskać od jądra informację, że została ona poprawnie wykryta. Nie ma to jednak miejsca zawsze, jest to tylko dodatkowa informacja.

Listing 4.1: Pozyskiwanie informacji od jądra

```
(Dla drukarek na porcie równoległym)
$ dmesg | grep -i print
parport0: Printer, Hewlett-Packard HP LaserJet 2100 Series

(Dla drukarek USB)
$ lsusb
(...)
Bus 001 Device 007: ID 03f0:1004 Hewlett-Packard DeskJet 970c/970cse
```

I

Instalowanie drukarki

W celu zainstalowania drukarki należy uruchomić przeglądarkę i udać się pod adres <http://localhost:631>. Pojawi się strona służąca do zarządzania systemem CUPS.

Kolejny krok to przejście do zakładki Administration i wpisanie loginu "root" oraz odpowiedniego hasła. Potem zostaniemy przeniesieni w tryb zarządzania serwerem wydruku. Następnym naszym krokiem będzie przejście do zakładki Add Printer, gdzie pojawi się nowy ekran pozwalający na podanie następujących parametrów:

- spooler name - krótka nazwa pod jaką drukarka będzie rozpoznawana w systemie. Nie powinna zawierać spacji ani znaków specjalnych. Na przykład, dla drukarki HP LaserJet 5P, można wybrać nazwę hpljet5p.
- location, opis miejsca, w którym faktycznie znajduje się drukarka (np. w pomieszczeniu o podanym numerze). Ułatwia to zarządzanie kilkoma drukarkami.
- description, krótki opis urządzenia, zwykle używa się tego pola do podania pełnej nazwy używanego sprzętu, np. "HP LaserJet 5P".

Następny ekran służy do wyboru punktu, w którym drukarka nasłuchuje. Będzie można je wybrać spośród listy. W poniższej tabeli wypisujemy tylko kilka urządzeń, nie jest to ich kompletna lista.

Urządzenie	Opis
AppSocket/HP JetDirect	Umożliwia komunikację poprzez gniazdo HP JetDirect. Bardzo niewiele drukarek z tego korzysta.
Internet Printing Protocol (IPP or HTTP)	Służy do lokalizowania zdalnych drukarek poprzez protokół IPP, zarówno bezpośrednio (IPP) jak i za pośrednictwem HTTP.
LPD/LPR Host or Printer	Dla drukarek sieciowych podłączonych do serwerów LPD/LPR.
Parallel Port #1	Wybieramy, gdy drukarka jest podłączona do lokalnego portu równoległego (LPT).
USB Printer #1	Wybieramy, gdy drukarka jest podłączona do lokalnego portu USB.

Jeśli instalujemy zdalną drukarkę zostaniemy zapytani o jej adres URI.

- Nazwy drukarek serwera LPD mają składnię lpd://hostname/queue.
- Nazwy dla HP JetDirect składnię socket://hostname
- Nazwy drukarek IPP mają składnię http://hostname:631/ipp/queue lub ipp://hostname/ipp/queue.

Następnie musimy wybrać producenta, model oraz konkretny numer naszej drukarki. Wiele drukarek może korzystać z tych samych sterowników. Można albo wybrać jeden z domyślnych albo poszukać na stronie LinuxPrinting.org bardziej odpowiedniego sterownika. Sterownik można zresztą z łatwością później zmienić.

Po wybraniu sterownika CUPS poinformuje o udanym dodaniu drukarki do systemu. Można teraz przejść do strony zarządzania drukarkami i wybrać opcję Configure Printer w celu zmiany jej ustawień.

Testowanie i rekonfigurowanie drukarki

W celu sprawdzenia czy drukarka działa prawidłowo należy przejść do strony administracji i wybrać tam opcję Print Test Page, która wydrukuje stronę testową.

Jeśli drukarka nie działa prawidłowo należy wybrać opcję Modify Printer i poprawić jej konfigurację. Pojawi się ta sama seria pól konfiguracyjnych do wypełnienia co wcześniej, z tym, że tym razem jako domyślna będzie wyświetlana obecna konfiguracja.

Jeśli naprawdę nie wiadomo czemu drukarka nie działa, cennych wskazówek może dostarczyć przeglądanie pliku /var/log/cups/error_log. W kolejnym przykładzie zaistnieje błąd związany z prawami dostępu spowodowany złym wpisem w polu Allow pliku /etc/cups/cupsd.conf.

Listing 4.2: Przeglądanie błędów CUPS

```
# tail /var/log/cups/error_log
(...)
E [11/Jun/2005:10:23:28 +0200] [Job 102] Unable to get printer status (client-error-forbidden)
```

Wybór najlepszego sterownika

Istnieje wiele różnych sterowników. Aby odnaleźć ten, który zapewni najlepszą wydajność dla danej drukarki należy udać się na stronę LinuxPrinting Printer List i wybrać tam z listy odpowiedniego producenta i model sprzętu. Na przykład dla drukarki HP LaserJet 5P autorzy strony zalecają użycie sterownika ljjet4.

Pobrany ze strony plik PPD należy umieścić w katalogu /usr/share/cups/model a następnie uruchomić skrypt startowy /etc/init.d/cupsd restart jako root, co umożliwi korzystanie z niego poprzez stronę administracyjną, a następnie skonfigurować sprzęt do pracy z nim podobnie jak to opisujemy powyżej.

5. Używanie specjalnych sterowników

Wprowadzenie

Niektóre drukarki wymagają specjalnych sterowników w celu uaktywnienia specjalnych właściwości jakie posiadają, a które po przeprowadzeniu konfiguracji tak jak opisano to powyżej nie będą działać. W tym rozdziale opiszemy kilka konkretnych drukarek oraz sposoby ich konfigurowania tak, aby mogły korzystać z pełni swoich możliwości.

Omówimy następujące sterowniki:

- Sterownik Gimp-Print
- Sterownik HPLIP (zastępujący HPOJ i HPIJS)
- Sterownik PNM2PPA

Gimp-Print Driver

Otwarte sterowniki gimp-print zapewniają wysoką jakość wydruku w wielu drukarkach firm Canon, Epson, HP, Lexmark, Sony, Olympus i drukarkach PCL współpracujących z CUPS, ghostscript, GIMP-em i wieloma innymi aplikacjami.

Drzewo Portage zawiera ebuild instalujący sterowniki gimp-print. Do ich zainstalowania wystarczy zatem użycie polecenia emerge. Warto zauważyć, że ebuild ten posiada kilka wartościowych flag USE (jak np. cups i ppds). Co najmniej te dwie flagi muszą być włączone!

Listing 5.1: Instalowanie sterownika gimp-print

```
# emerge gimp-print
```

Kiedy proces instalacji się zakończy sterowniki gimp-print będą gotowe do wybrania ze strony administracyjnej CUPS.

HPLIP Driver

[Projekt HPLIP](#) to całkowicie nowe podejście do sterownika hpijs. Sterownik ten posiada również obsługę wielu skanerów oraz innych urządzeń. Aby można było drukować za jego pomocą musi on mieć włączoną flagę USE ppds.

Listing 5.2: Instalowanie sterowników hplip

```
Należy się upewnić, że flaga USE ppds jest włączona
# emerge hplip
```

```
These are the packages that I would merge, in order:
```

```
Calculating dependencies ...done!
```

```
[ebuild N      ] net-print/hplip-0.9.5  +X +cups +foomaticdb +ppds -qt -scanner -snmp +usb 9,255 kB
```

```
Total size of downloads: 9,255 kB
```

```
# emerge hplip
# /etc/init.d/hplip start
# /etc/init.d/cupsd restart
# rc-update add hplip default
```

Po zakończeniu procesu instalacji sterowniki będą dostępne do wyboru z poziomu interfejsu www CUPS.

Sterownik PNM2PPA

PPA to technologia firmy HP, która skupia się na przesyłaniu niskopoziomowych sygnałów do systemu zamiast do drukarki, co powoduje, że drukarka staje się tańsza, ale również konsumuje więcej zasobów.

Jeśli na stronie [LinuxPrinting](#) znajduje się informacja, że sterownik [pnm2ppa](#) jest najlepszym możliwym wyborem to należy zainstalować filtry pnmm2ppa.

Listing 5.3: Instalowanie filtra pnmm2ppa

```
# emerge pnmm2ppa
```

Po zainstalowaniu pobieramy plik PPD dla naszej drukarki ze strony [LinuxPrinting](#) i umieszczamy go w katalogu /usr/share/cups/model. Następnie konfigurujemy drukarkę tak jak opisano to powyżej.

6. Drukowanie przy współpracy z systemami Microsoft Windows

Konfigurowanie klienta IPP w systemie Windows

Microsoft Windows posiada obsługę IPP (użytkownicy Windows 9x oraz ME muszą [zainstalować](#) ją osobno). Aby zainstalować drukarkę podłączoną do komputera z Linuksem należy uruchomić program konfiguracyjny służący do dodawania drukarek i wybrać w nim opcję drukarka sieciowa. Przy pytaniu o URI należy podać odpowiedni adres o składni http://hostname:631/printers/queue.

Należy się upewnić, że komputer jest w stanie [połączyć się](#) z drukarką!

Konfiguracja klienta Windows do pracy z drukarką współdzieloną za pomocą Samby

W celu korzystania ze współdzielonej drukarki w sieci SMB-CIFS należy zainstalować i skonfigurować system SAMBA. Czynności na to się składające nie są tematem tego dokumentu, ale postaramy się pokróćce omówić proces konfiguracji Samby dla współdzielonych drukarek.

Otwieramy plik /etc/samba/smb.conf naszym ulubionym edytorem i dodajemy do niego odpowiedni akapit:

Listing 6.1: Dodawanie akapitu [printers]

```
[printers]
comment      = All printers
path         = /var/spool/samba
browseable   = no
guest ok     = no
writable     = no
printable    = yes
public       = yes
printer name = hpljet5p
```

Następnie przewijamy plik smb.conf aż do znalezienia akapitu [global]. Wewnątrz niego znajdujemy wpisy printcap name i printing i

ustawiamy każdy z nich na cups:

Listing 6.2: Zmiana akapitu [globals]

```
[global]
(...)

printcap name = cups
printing      = cups
```

Należy się upewnić, że komputer obsługuje sterowniki PCL, a następnie ponownie uruchomić usługę smb z nowymi ustawieniami.

Konfigurowanie klienta Linuksowego do pracy z windowsowym serwerem druku

Po pierwsze należy się upewnić, że Windows udostępnia drukarkę.

Następnie na stronie będącej interfejsem administracyjnym dla CUPS należy skonfigurować drukarkę tak jak opisano to wcześniej. Powinna pojawić się informacja o tym, że CUPS dodał nową drukarkę o nazwie Windows Printer via SAMBA. Należy ją wybrać i użyć składni smb://username:password@workgroup/server/printename lub smb://server/printename jako jej URI.

7. Programy związane z drukowaniem

Wprowadzenie

Istnieje wiele programów ułatwiających konfigurację drukarki, korzystanie z dodatkowych filtrów druku, czy dodających różne funkcje dotyczące drukowania. W tym rozdziale omówimy kilka z nich. Na pewno nie są to wszystkie programy tego typu, nie omawiamy ich też zbyt szczegółowo.

Gtk-LP - Program na GTK służący do konfiguracji drukarki

Dzięki Gtk-LP można zainstalować, konfigurować oraz modyfikować ustawienia drukarki. Program korzysta z CUPS i posiada wszystkie jego możliwości. Warto go zainstalować jeśli interfejs www dla CUPS nie wydaje się dobrym wyborem lub gdy potrzebna jest osobna aplikacja do codziennego wydruku dokumentów.

Instaluje się go emergując pakiet gtklp:

Listing 7.1: Instalowanie Gtk-LP

```
# emerge gtklp
```

8. Problemy

Error: Unable to convert file 0 to printable format

Jeśli pojawią się problemy z drukowaniem, a w pliku /var/log/cups/error_log pojawia się taki wpis:

Listing 8.1: Błąd

```
Unable to convert file 0 to printable format
```

Oznacza to, że należy ponownie zainstalować pakiet ghostscript-esp z flagą USE cups. Można dodać ją zarówno w pliku /etc/make.conf jak i w /etc/portage/package.use:

Listing 8.2: Włączanie flagi USE cups

```
# echo "app-text/ghostscript-esp cups" >> /etc/portage/package.use
```

Później należy przebudować pakiet ghostscript-esp i ponownie uruchomić CUPS.

F. Administrowanie Gentoo

1. Bezpieczeństwo systemu

1.a. Fizyczne bezpieczeństwo

Nieważne jak wiele rozwiązań związanych z zabezpieczeniami się zaimplementuje, wszystkie mogą zostać łatwo ominięte przez

włamywacza mającego fizyczny dostęp do komputera. Należy zatem zastosować serię środków uniemożliwiających taki dostęp. Na przykład zamknięcie komputera w szafie uniemożliwi włamywaczowi jego odłączenie od prądu, a zamknięcie obudowy na klucz uniemożliwi kradzież twardego dysku. Aby uniemożliwić uruchomienie systemu z innego dysku i obejście wszystkich zabezpieczeń logowania oraz tych związanych z prawami dostępu do plików należy ustawić twardy dysk jako pierwsze i jedyne urządzenie uruchamialne w BIOS-ie oraz zabezpieczyć dostęp do BIOS-u hasłem. Kolejną ważną czynnością jest ustawienie hasła dla LILO lub GRUB-a, co uniemożliwi zalogowanie się z nich do systemu w trybie single-user i w konsekwencji uzyskanie pełnego do niego dostępu. Więcej informacji na ten temat znajduje się w rozdziałach Ustawianie hasła dla GRUB i Ustawianie hasła dla LILO.

1.b. Planowanie usług i demonów

Na samym początku należy podjąć decyzję, które usługi będą uruchomione na danej maszynie. Pozwoli to na lepsze zaplanowanie schematu podziału dysku na partycje. Oczywiście nie jest to konieczne na sprzętzie przeznaczonym na stację roboczą lub dedykowany firewall. W takim przypadku nie powinny być uruchomione żadne usługi, no może poza demonem sshd.

Utworzenie listy z aktualnymi numerami wersji uruchomionych programów ułatwi ich aktualnianie gdy pojawią się informacje o błędach w jednym nich.

1.c. Schematy partycjonowania

Zasady partycjonowania:

- Wszystkie katalogi w których użytkownicy będą mieli możliwość zapisywania plików (np. /home, lub /tmp) powinny znajdować się na osobnych partycjach używających quota. Zmniejsza to ryzyko zapełnienia całego dostępnego na dysku miejsca. Portage używa /var/tmp do komplikowania plików, więc ta partycja powinna być stosunkowo duża.
- Każdy katalog przeznaczony do instalowania oprogramowania spoza dystrybucji powinien znajdować się na osobnej partycji. Zgodnie ze standardem hierarchii plików katalogi te to /opt lub /usr/local. Jeśli są na osobnych partycjach nie zostaną wyczyszczone przy ponownej instalacji systemu.
- Dla jeszcze większego bezpieczeństwa nie zmieniające się dane mogą zostać zapisane na partycji podmontowanej tylko do odczytu. Jeśli administrator jest typem paranoika może używać do tego celu nawet dysków CD-ROM. :]

1.d. Użytkownik root

Użytkownik root jest najważniejszą osobą w systemie. Jego konto nie powinno być używane tylko wtedy gdy jest to niezbędne. Jeśli włamywacz uzyska dostęp do tego konta to konieczna będzie ponowna instalacja całego systemu.

Zasady postępowania z kontem roota:

- Do codziennej pracy należy utworzyć zwykłe konto użytkownika. Jeśli wymagane jest częste przełączanie się na roota warto dodać to konto do grupy 'wheel'. Umożliwi to korzystanie z polecenia su, które znacznie upraszcza procedurę przełączania się na konto superużytkownika.
- Nigdy nie należy uruchamiać serwera X lub jakiekolwiek aplikacji użytkownika z przywilejami roota. Konto roota powinno być używane tylko wtedy, gdy jest to niezbędne. Jeśli uruchamiana z normalnymi prawami aplikacja zawiera błąd to włamywacz uzyska dostęp jedynie do zwykłego konta, jeśli uruchomi się ją na prawach roota lub z dostępu do całego systemu.
- Należy używać bezwzględnych ścieżek do programów kiedy jest się zalogowanym jako root (lub zawsze używać su -, które zastępuje zmienne środowiskowe użytkownika tymi należącymi do roota, zmienna PATH roota powinna zawierać jedynie chronione katalogi jak /bin i /sbin). Istnieje możliwość oszukania roota i zmuszenia go do uruchomienia innej aplikacji niż zamierzał. Jeśli PATH superużytkownika jest chroniona lub root zawsze używa absolutnych ścieżek nie ma możliwości by to się stało.
- Jeśli użytkownik potrzebuje wpisać jako root tylko kilka wybranych komend powinno się rozważyć użycie sudo zamiast przełogowywania się na konto roota. Oczywiście należy zwracać baczną uwagę na to komu daje się dostęp do tego polecenia.
- Nigdy nie należy zostawiać otwartego terminala z zalogowanym do niego rootem.

Gentoo posiada kilka domyślnych zabezpieczeń przed zwykłymi użytkownikami, którzy chcą używać polecenia su do przełączania się na konto roota. Domyślne ustawienie PAM umożliwia korzystanie z takiego ułatwienia jedynie tym, którzy są członkami grupy 'wheel'.

1.e. Polityka bezpieczeństwa

Istnieje kilka powodów dla których warto naszkicować politykę bezpieczeństwa dla swojego systemu i sieci.

- Dobra polityka bezpieczeństwa powinna dotyczyć całego systemu, a nie być jedynie zbieranicą różnych jego właściwości. Na przykład administrator, który nie przemyślał polityki bezpieczeństwa mógłby wyłączyć telnet, ponieważ umożliwia przesyłanie niezaszyfrowanych haseł, ale pozostawić dostęp przez FTP, który posiada takie same słabości. Dobra polityka bezpieczeństwa umożliwia rozpoznanie wartych i niewartych zastosowania środków bezpieczeństwa.
- W celu diagnozowania problemów, przeprowadzania audytu czy wykrywania włamania administrator jest zmuszony do przechwytywania ruchu w sieci, przeglądania historii logowania i poleceń poszczególnych użytkowników oraz kontroli ich katalogów domowych. Jeśli nie poinformuje o tym użytkowników może mieć poważne problemy z prawem, taka działalność może zostać uznana za naruszenie ich prywatności i wpadźić go w spore kłopoty.
- Bardzo często dużym zagrożeniem dla systemu są przejęte konta użytkowników. Jeśli nie wyjaśniono użytkownikom jak mają postępować (np. nie zabroni się im przyklejania listy haseł na kartkach do biurka) zabezpieczanie ich kont będzie bezcelowe.
- Dobrze udokumentowany rozkład sieci i systemu pomoże osobom badającym system po włamaniu w zidentyfikowaniu zarówno osoby włamywacza jak i sposobu w jaki dokonał włamania. Polityka bezpieczeństwa powinna zawierać również informację, że cały system jest prywatną siecią i każda nieautoryzowana próba dostępu jest zabroniona, co później może ułatwić dochodzenie swoich praw po namierzeniu włamywacza.

Jak widać dobra polityka bezpieczeństwa jest bardzo przydatnym narzędziem.

Polityka powinna być pojedynczym dokumentem lub ich zbiorem, który opisuje całą sieć oraz usługi przez nią zapewniane, dozwolone oraz zakazane czynności, zwyczaje obowiązujące w sieci i tak dalej. Wszyscy użytkownicy powinni zapoznać się z tym dokumentem oraz być powiadomiani o wszelkich dokonywanych w nim zmianach, które powinno się nanosić, aby dokument był aktualny. Warto pomóc użytkownikom w zrozumieniu całej polityki oraz wyjaśnić dlaczego należy jej przestrzegać, a także jasno zaznaczyć jakie poniosą konsekwencje jeśli postąpią wbrew niej (dokument powinien zawierać wszystkie te informacje). Należy przypominać im o tym co najmniej raz w roku, ponieważ polityka może się zmieniać (a także, żeby nie zapomnieli, że taki dokument istnieje).

Uwaga: Dokumenty te powinny być zrozumiałe, ale warto również zachować maksymalną precyzję w każdym z poruszanych tematów.

Każda polityka bezpieczeństwa powinna (co najmniej) poruszać następujące tematy:

- Właściwe postępowanie
 - Wygaszanie ekranu.
- Sposób postępowania z hasłami.
- Zasady pobierania i instalowania programów.
- Informację o tym, że działania użytkowników są monitorowane.
- Informacje o oprogramowaniu antywirusowym.
- Sposób postępowania z krytycznymi informacjami.
 - Czyszczenie biurka oraz zamykanie na klucz informacji utajnionych.
- Wyłączanie komputera przed wyjściem.
- Używanie szyfrowania.
- Sposób postępowania z kluczami zaufanych współpracowników.
- Sposób postępowania z poufnymi danymi podczas podróży.
- Sposób postępowania ze sprzętem komputerowym podczas podróży.
 - Postępowanie z laptopami w czasie podróży i pobytu w hotelach.

Różni użytkownicy będą zapewne posiadać różny zakres dostępu. W polityce warto uwzględnić wszystkie przypadki.

Polityka bezpieczeństwa ma tendencję do nadmiernego rozrastania się, co może prowadzić do przeoczenia najważniejszych informacji. Ponadto wersja dokumentu dla pracowników IT może zawierać poufne informacje, które nie powinny być znane wszystkim użytkownikom. W związku z tym warto podzielić całość na serię mniejszych dokumentów, np. dotyczących osobno: haseł, poczty, zdalnego dostępu itd.

Przykładowe polityki bezpieczeństwa można znaleźć na stronach [The SANS Security Policy Project](#). Dla małych sieci, które nie potrzebują tak rozbudowanych opracowań powinny wystarczyć polityki takie jak ta pod adresem [Site Security Handbook](#).

2. Uszczelnianie systemu

2.a. Flagi USE

Plik make.conf zawiera zdefiniowane przez użytkownika flagi USE, a w pliku /etc/make.profile/make.defaults znajdują się te wybrane jako domyślne dla Gentoo. Ten przewodnik nawiązuje do flag pam (Pluggable Authentication Modules), tcpd (TCP wrappers) i ssl (Secure Socket Layer). Wszystkie są włączane w domyślnej konfiguracji.

2.b. Ochrona hasłem w GRUB

Istnieją dwa sposoby zabezpieczenia tego bootloadera hasłem. Pierwszy przy użyciu otwartego tekstu, a drugi z hasłem zaszyfrowanym przy pomocy algorytmu md5+salt.

Listing 1: /boot/grub/grub.conf

```
timeout 5
password changeme
```

Taki wpis ustala hasło na ciąg znaków zmień_mnie. Jeśli hasło nie zostanie wpisane przy starcie systemu, GRUB użyje domyślnych ustawień.

Przed ustawieniem hasła w postaci md5 trzeba je zaszyfrować do formatu takiego jak ten używany w pliku /etc/shadow. Więcej informacji można znaleźć w dokumencie man crypt. Zaszyfrowane hasło zmień_mnie ma taką postać: 4ff4d55e97d977e3b9d04b5418426403.

Zaszyfrowania hasła można dokonać bezpośrednio z powłoki GRUB-a:

Listing 2: md5crypt w powloce grub

```
#/sbin/grub

GRUB version 0.92 (640K lower / 3072K upper memory)

[ Minimal BASH-like line editing is supported. For the first word, TAB lists
possible command completions. Anywhere else TAB lists the possible
completions of a device/filename. ]

grub> md5crypt

Password: *****
(Typed changeme at the prompt)
Encrypted: $1$T7/dgdIJ$dJM.n2wZ8RG.oEiIowJUs.

grub> quit
```

Następnie należy skopiować zaszyfrowane hasło i wkleić je do pliku /boot/grub/grub.conf.

Listing 3: /boot/grub/grub.conf

```
timeout 5
password --md5 $1$T7/dgdIJ$dJM.n2wZ8RG.oEiIowJUs.
```

Pięciosekundowe opóźnienie jest przydatne przy zdalnej administracji systemem, kiedy chcemy mieć możliwość uruchomienia systemu bez bezpośredniego do niego dostępu. Więcej informacji o hasłach w GRUB można uzyskać wpisując polecenie info grub.

2.c. Ochrona hasłem w LILO.

LILO również posiada dwie metody postępowania z hasłami: globalną oraz osobną dla każdego obrazu, niestety nie ma możliwości ich zaszyfrowania.

Globalne hasło ustawia się na samej górze pliku konfiguracyjnego, ma ono zastosowanie dla każdego obrazu jądra:

Listing 4: /etc/lilo.conf

```
password=changeme
restricted
delay=3
```

Poniżej znajduje się przykład ustawienia hasła dla określonego obrazu:

Listing 5: /etc/lilo.conf

```
image=/boot/bzImage
      read-only
      password=changeme
      restricted
```

Jeśli nie dopiszemy opcji restricted będziemy za każdym razem pytani o hasło.

Żeby zastosować zmiany należy uruchomić /sbin/lilo.

2.d. Ograniczanie dostępu do konsoli

W pliku /etc/securetty można skonfigurować te urządzenia tty (terminale), na których będzie mógł się logować root.

Zalecamy zakomentowanie wszystkich linii poza vc/1 jeśli używa się devfs i wszystkich linii poza tty1 jeśli korzysta się z udev. Spowoduje to, że root będzie mógł być zalogowany tylko raz i tylko na jednym terminalu.

Uwaga: Użytkownicy z grupy 'wheel' wciąż będą mogli stawać się rootem przy pomocy polecenia su - na innych terminalach.

Listing 6: /etc/securetty

```
(Dla devfs)
vc/1
(Dla udev)
tty1
```

3. Logowanie

3.a. Wprowadzenie

Aby wychwycić wszystkie ostrzeżenia i błędy wskazujące na próby ataku należy włączyć dodatkowe usługi logowania. Włamywacze zwykle skanują i wypróbowują system przed atakiem.

Ważną rzeczą jest również to, by logi były łatwe w zarządzaniu i przeglądaniu. Gentoo podczas instalacji umożliwia wybranie jednego z trzech programów logujących.

3.b. Logowanie za pomocą syslogd

Syslogd jest najpopularniejszym programem logującym dla systemów Linux i Unix. Nie daje możliwości rotowania logów. Można to sobie jednak zrekompensować używając programu /usr/sbin/logrotate, warto dopisać go np. w demonie cron (plik konfiguracyjny logrotate to /etc/logrotate.conf). Częstotliwość rotacji logów będzie uzależniona od obciążenia systemu.

Poniżej znajduje się standardowy plik syslog.conf do którego wprowadzono kilka pozytywnych zmian. Odkomentowano linie dotyczące crona i tty oraz dodano opcję wysyłania logów na zdalny serwer. Źeby jeszcze bardziej zwiększyć bezpieczeństwo, można skonfigurować program tak, aby zapisywał logi w dwóch różnych miejscach.

Listing 1: /etc/syslog.conf

```
# /etc/syslog.conf      Configuration file for syslogd.
#
#                                     For more information see syslog.conf(5)
# manpage.
# This is from Debian, we are using it for now
# Daniel Robbins, 5/15/99

#
# First some standard logfiles.  Log by facility.
#


auth,authpriv.*          /var/log/auth.log
*.*,auth,authpriv.none   -/var/log/syslog
cron.*                   /var/log/cron.log
daemon.*                 -/var/log/daemon.log
kern.*                   -/var/log/kern.log
lpr.*                     -/var/log/lpr.log
mail.*                   /var/log/mail.log
user.*                   -/var/log/user.log
uucp.*                   -/var/log/uucp.log
local6.debug             /var/log/imapd.log
#
# Logging for the mail system. Split it up so that
# it is easy to write scripts to parse these files.
#
mail.info                -/var/log/mail.info
mail.warn                -/var/log/mail.warn
mail.err                 /var/log/mail.err
#
# Logging for INN news system
#
news.crit                /var/log/news/news.crit
news.err                  /var/log/news/news.err
news.notice               -/var/log/news/news.notice
#
# Some `catch-all' logfiles.
#
*=debug; \
    auth,authpriv.none; \
    news.none;mail.none   -/var/log/debug
*=info;*=notice;*=warn; \
    auth,authpriv.none; \
    cron,daemon.none; \
    mail,news.none        -/var/log/messages

#
# Emergencies and alerts are sent to everybody logged in.
```

```

#
*. emerg          *
*. =alert         *
#
# I like to have messages displayed on the console, but only on a virtual
# console I usually leave idle.
#
daemon,mail.*;\*
    news.=crit;news.=err;news.=notice; \
    *.=debug;*.=info; \
    *.=notice;*.=warn      /dev/tty8

#Setup a remote logging server
*.*               @logserver

# The named pipe /dev/xconsole is for the `xconsole' utility. To use it,
# you must invoke `xconsole' with the `-file' option:
#
#     $ xconsole -file /dev/xconsole [...]
#
# NOTE: adjust the list below, or you'll go crazy if you have a reasonably
#       busy site..
#
#daemon.*,mail.*;\*
#    news.crit;news.err;news.notice; \
#    *.=debug;*.=info; \
#    *.=notice;*.=warn      |/dev/xconsole

local2.*          --/var/log/ppp.log

```

Włamywacze zwykle próbują zatrzeć po sobie ślady poprzez wyedytowanie lub skasowanie logów. Można im to utrudnić ustawiając zdalne logowanie na innych komputerach.Więcej informacji o syslogd znajduje się w man syslog.

3.c. Metalog

Metalog Franka Dennisa nie ma możliwości logowania na zdalnym serwerze. Ma za to sporą przewagę jeśli chodzi o elastyczność i wydajność logowania. Może logować nazwę programu, priorytet i wiele innych informacji. Daje możliwość korzystania z wyrażeń regularnych oraz uruchamiania skryptów w przypadku wystąpienia w logu określonego ciągu znaków. Bardzo przydaje się to do błyskawicznego wykonywania niezbędnych działań w przypadku gdy dzieje się coś złego.

Zwykle wystarcza domyślna konfiguracja. Aby być powiadamianym e-mailem za każdym razem gdy wystąpi błąd przy wpisywaniu hasła można użyć jednego z następujących skryptów:

Dla postfix:

Listing 2: /usr/local/sbin/mail_pwd_failures.sh for postfix

```

#!/bin/sh
echo "$3" | mail -s "Warning (program : $2)" root

```

Dla qmail:

Listing 3: /usr/local/sbin/mail_pwd_failures.sh for qmail

```

#!/bin/sh
echo "To: root
Subject:Failure (Warning: $2)
$3
" | /var/qmail/bin/qmail-inject -f root

```

Aby skrypt zadziałał należy mu nadać prawa wykonywalności poprzez wpisanie /bin/chmod +x /usr/local/sbin/mail_pwd_failures.sh.

Następnie należy odkomentować linię "Password failures" w pliku /etc/metalog/metalog.conf:

Listing 4: /etc/metalog/metalog.conf

```
command = "/usr/local/sbin/mail_pwd_failures.sh"
```

3.d. Syslog-ng

Syslog-ng posiada te same funkcje co syslog i metalog, jednak z małymi różnicami. Może filtrować wiadomości zależnie od poziomu i zawartości (jak metalog), umożliwia zdalne logowanie (jak syslog), dobrze współpracuje z rozmaitymi plikami, tak jak syslogd (nawet ze strumieniami z Solarisa) potrafi wyświetlać logi na tty, wykonywać programy i może pełnić funkcję serwera logów. Jest najlepszym dostępnym programem tego typu i jakby tego było mało posiada jeszcze najszerze możliwości konfiguracyjne.

Poniżej znajduje się standardowy plik konfiguracyjny na który naniesiono kilka drobnych poprawek.

Listing 5: /etc/syslog-ng/syslog-ng.conf

```
options { chain_hostnames(off); sync(0); };

#source where to read log
source src { unix-stream("/dev/log"); internal(); };
source kernsrc { file("/proc/kmsg"); };

#define destinations
destination authlog { file("/var/log/auth.log"); };
destination syslog { file("/var/log/syslog"); };
destination cron { file("/var/log/cron.log"); };
destination daemon { file("/var/log/daemon.log"); };
destination kern { file("/var/log/kern.log"); };
destination lpr { file("/var/log/lpr.log"); };
destination user { file("/var/log/user.log"); };
destination mail { file("/var/log/mail.log"); };

destination mailinfo { file("/var/log/mail.info"); };
destination mailwarn { file("/var/log/mail.warn"); };
destination mailerr { file("/var/log/mail.err"); };

destination newscrit { file("/var/log/news/news.crit"); };
destination newserr { file("/var/log/news/news.err"); };
destination newsnotice { file("/var/log/news/news.notice"); };

destination debug { file("/var/log/debug"); };
destination messages { file("/var/log/messages"); };
destination console { usertty("root"); };
destination console_all { file("/dev/tty12"); };
destination xconsole { pipe("/dev/xconsole"); };

#create filters
filter f_authpriv { facility(auth, authpriv); };
filter f_syslog { not facility(authpriv, mail); };
filter f_cron { facility(cron); };
filter f_daemon { facility(daemon); };
filter f_kern { facility(kern); };
filter f_lpr { facility(lpr); };
filter f_mail { facility(mail); };
filter f_user { facility(user); };
filter f_debug { not facility(auth, authpriv, news, mail); };
filter f_messages { level(info..warn)
    and not facility(auth, authpriv, mail, news); };
filter f_emergency { level(emerg); };

filter f_info { level(info); };
filter f_notice { level(notice); };
filter f_warn { level(warn); };
filter f_crit { level(crit); };
filter f_err { level(err); };
filter f_failed { match("failed"); };
filter f_denied { match("denied"); };

#connect filter and destination
```

```

log { source(src); filter(f_authpriv); destination(authlog); };
log { source(src); filter(f_syslog); destination(syslog); };
log { source(src); filter(f_cron); destination(cron); };
log { source(src); filter(f_daemon); destination(daemon); };
log { source(kernsrc); filter(f_kern); destination(kern); };
log { source(src); filter(f_lpr); destination(lpr); };
log { source(src); filter(f_mail); destination(mail); };
log { source(src); filter(f_user); destination(user); };
log { source(src); filter(f_mail); filter(f_info); destination(mailinfo); };
log { source(src); filter(f_mail); filter(f_warn); destination(mailwarn); };
log { source(src); filter(f_mail); filter(f_err); destination(mailerr); };

log { source(src); filter(f_debug); destination(debug); };
log { source(src); filter(f_messages); destination(messages); };
log { source(src); filter(f_emergency); destination(console); };

#default log
log { source(src); destination(console_all); };

```

Konfiguracja syslog-ng jest dość łatwa, ale z powodu ogromnych rozmiarów całego pliku łatwo przegapić coś ważnego. Autor wciąż obiecuje dodatkowe opcje, jak na przykład szyfrowanie, uwierzytelnianie, kompresja czy kontrola MAC (Mandatory Access Control). Z tymi opcjami program będzie idealnym narzędziem do logowania ruchu sieciowego, bo uniemożliwi włamywaczom szpiegowanie logów.

Ogromną zaletą syslog-ng jest również to, że nie musi być uruchamiany z prawami roota.

3.e. Analiza logów przy pomocy logcheck.

Oczywiście uniemożliwienie dostępu do logów to tylko połowiczne zwycięstwo. Aplikacje takie jak logcheck znacznie ułatwiają ich codzienną analizę. Logcheck to skrypt powiązany z plikiem binarnym o nazwie logtail, który jest zwykle uruchamiany z demona cron i dzięki zestawowi prostych reguł sprawdza logi pod kątem występowania podejrzanych wyrażeń. Jeśli coś takiego wystąpi wysyła e-mail na skrzynkę roota.

Logcheck i logtail są częścią pakietu app-admin/logsentry.

Do filtrowania ważnych wpisów w logach logcheck używa informacji z czterech plików. Te pliki to logcheck.hacking, który zawiera znane wiadomości opisujące atak, logcheck.violations, który zawiera wzory naruszeń bezpieczeństwa, logcheck.violations.ignore z informacjami pozwalającymi na zdefiniowanie często występujących, omyłkowo branych za ataki wpisów w logach oraz logcheck.ignore, który pozwala na ich zignorowanie.

4. Montowanie partycji

4.a. Montowanie partycji

Podczas montowania partycji ext2, ext3 lub reiserfs mamy do dyspozycji szereg, ważnych z punktu widzenia bezpieczeństwa, opcji. Są to:

- nosuid - ignoruje wszystkie bity SUID, przez co wszystkie pliki z nimi będą traktowane jakby ich nie miały.
- noexec - uniemożliwia uruchomienie jakiegokolwiek programu z tej partycji.
- nodev - ignoruje urządzenia.

Niestety te ustawienia można w łatwy sposób obejść poprzez użycie pośredniej ścieżki. Mimo wszystko jednak ustawienie /tmp na noexec uniemożliwi wykonanie większości exploitów zaprojektowanych tak, aby zostały wykonane właśnie tam.

Listing 1: /etc/fstab

```
/dev/sda1 /boot ext2 noauto,noatime 1 1
/dev/sda2 none swap sw 0 0
/dev/sda3 / reiserfs notail,noatime 0 0
/dev/sda4 /tmp reiserfs notail,noatime,nodev,nosuid,noexec 0 0
/dev/sda5 /var reiserfs notail,noatime,nodev 0 0
/dev/sda6 /home reiserfs notail,noatime,nodev,nosuid 0 0
/dev/sda7 /usr reiserfs notail,noatime,nodev,ro 0 0
/dev/cdroms/cdrom0 /mnt/cdrom iso9660 noauto,ro 0 0
proc /proc proc defaults 0 0
```

Ostrzeżenie: Przelączanie /tmp w tryb noexec uniemożliwi prawidłowe wykonanie niektórych skryptów.

Uwaga: Więcej informacji o quotach znajduje się w sekcji Quota.

Uwaga: Osobiście nie ustawiam /var jako noexec i nosuid, nawet jeśli żadne pliki nie są normalnie stamtąd wykonywane. Powodem tego jest fakt, że w /var/qmail mam zainstalowany qmail, który musi mieć możliwość uruchomienia się z bitem SUID. Partycję /usr przestawiam na read-only, ponieważ nigdy nic tam nie zapisuję, chyba, że akurat aktualniam Gentoo. W takich wypadkach przemontowuję tą partycję w tryb do zapisu, aktualniam system i znów przestawiam ją na 'tylko do odczytu'.

Uwaga: Nawet jeśli nie używa się qmail, Gentoo wciąż będzie potrzebowało możliwości uruchamiania programów w /var/tmp ponieważ są tam tworzone pliki ebuild. Można zmienić tą ścieżkę jeśli koniecznie chce się montować /var jako noexec.

5. Limitowanie grup i użytkowników

5.a. /etc/security/limits.conf

Kontrola użycia zasobów jest bardzo efektywnym sposobem zapobiegania lokalnym atakom typu "Denial of Service". Możemy też przy okazji również nałożyć restrykcje dla maksymalnej dopuszczalnej ilości zalogowań dla użytkowników oraz grup. Nie wolno jednak ustawiać zbyt restrykcyjnych limitów, gdyż może to zakłócić prawidłowe działanie systemu. Każdy wpis należy sprawdzić kilkukrotnie zanim się go zacznie używać.

Listing 1: /etc/security/limits.conf

```
*      soft core 0
*      hard core 0
*      hard nproc 15
*      hard rss 10000
*      -      maxlogins 2
@dev hard core 100000
@dev soft nproc 20
@dev hard nproc 35
@dev -      maxlogins 10
```

Zamiast ustawiania wartości nproc lub maxlogins na 0 należy rozważyć całkowite usunięcie takiego użytkownika. W powyższym przykładzie ustawiliśmy grupie dev ilość procesów, plików core oraz wartość maxlogins. Reszta to wartości domyślne.

Uwaga: /etc/security/limits.conf jest częścią pakietu PAM i ma zastosowanie wyłącznie dla pakietów go używających.

5.b. /etc/limits

Plik /etc/limits jest bardzo podobny do pliku /etc/security/limits.conf. Jedyne różnice to format pliku oraz to, że ma on zastosowanie dla użytkowników, a nie dla grup. Jego przykładowa konfiguracja wygląda tak:

Listing 2: /etc/limits

```
*      L2 C0 U15 R10000
kn     L10 C100000 U35
```

Definiujemy tu domyślne ustawienia dla użytkownika kn. Plik ten jest częścią pakietu sys-apps/shadow. Nie ma tu sensu ustawiać czegokolwiek jeśli nie skonfigurujesz prawidłowo PAM lub całkiem ją wyłączysz w pliku make.conf.

5.c. Quota

Ostrzeżenie: Należy upewnić się, że używany system plików posiada wsparcie dla quota. Żeby używać ich na ReiserFS trzeba nałożyć na kernel łatkę Namesys. Narzędzia użytkownika są dostępne na stronach Projektu DiskQuota dla Linux. Spowoduje to, że quoty będą działać, ale nie oznacza, że nie przysporzą żadnych problemów.

Umieszczenie quoty na systemie plików nałoży ograniczenie w użyciu przestrzeni dyskowej zarówno dla grupy jak i dla użytkowników. Quoty włącza się w jądrze, a następnie dodaje je do punktu montowania w pliku /etc/fstab. Opcja w konfiguracji kernela, które je włącza to File systems->Quota support. Należy zaznaczyć wymagane opcje, przebudować jądro, a następnie przy jego pomocy uruchomić system.

Zaczniemy od zainstalowania quota przy pomocy polecenia emerge quota. Następnie zmodyfikujemy plik /etc/fstab dodając tam usrquota i grpquota dla partycji, na których chcemy ograniczać użycie dysku.

Listing 3: /etc/fstab

```
/dev/sda1 /boot ext2 noauto,noatime 1 1
/dev/sda2 none swap sw 0 0
/dev/sda3 / reiserfs notail,noatime 0 0
/dev/sda4 /tmp ext3 noatime,nodev,nosuid,noexec,usrquota,grpquota 0 0
/dev/sda5 /var ext3 noatime,nodev,usrquota,grpquota 0 0
```

```
/dev/sda6 /home ext3 noatime,nodev,nosuid,usrquota,grpquota 0 0
/dev/sda7 /usr reiserfs notail,noatime,nodev,ro 0 0
/dev/cdroms/cdrom0 /mnt/cdrom iso9660 noauto,ro 0 0
proc /proc proc defaults 0 0
```

Na każdej partycji na której uaktywniono quota należy stworzyć pliki quota (quota.user i quota.group) oraz umieścić je w katalogu głównym każdej partycji.

Listing 4: Tworzenie plików quota

```
# touch /tmp/aquota.user
# touch /tmp/aquota.group
# chmod 600 /tmp/aquota.user
# chmod 600 /tmp/aquota.group
```

Trzeba powtórzyć te czynności osobno dla każdej partycji na której chce się używać quota. Po skonfigurowaniu wszystkich plików dodajemy skrypt quota do runlevelu boot.

Listing 5: Dodawanie skryptu quota do poziomu uruchamiania 'boot'

```
# rc-update add quota boot
```

Następnie konfigurujemy system tak, żeby sprawdzał quota raz w tygodniu. Robimy to dodając następującą linię do pliku /etc/crontab:

Listing 6: Dodawanie sprawdzania quota w crontabie

```
0 3 * * 0 /usr/sbin/quotacheck -avug.
```

Po ponowym uruchomieniu komputera przychodzi pora na skonfigurowanie quota dla poszczególnych użytkowników i grup. Polecenie edquota -u kn otworzy edytor zdefiniowany zmienną \$EDITOR (domyślnie jest to nano) i pozwoli wyedytować quota dla użytkownika kn. Polecenie edquota -g robi to samo dla grup.

Listing 7: Ustawianie quota dla użytkownika kn

```
Quotas for user kn:
/dev/sda4: blocks in use: 2594, limits (soft = 5000, hard = 6500)
          inodes in use: 356, limits (soft = 1000, hard = 1500)
```

Aby uzyskać więcej informacji należy zapoznać się z dokumentami man edquota i [Mini Quota HOWTO](#).

5.d. /etc/login.defs

Jeśli w polityce bezpieczeństwa założono, że użytkownicy powinni co tydzień zmieniać hasło należy zmienić wartość zmiennej PASS_MAX_DAYS na 14 oraz zmiennej PASS_WARN_AGE na 7. Zalecamy częste zmianianie haseł ponieważ metodami brute force można złamać każde z nich, jeśli tylko ma się dość dużo czasu. Ponadto polecamy ustawienie zmiennej LOG_OK_LOGINS na 'yes'.

5.e. /etc/login.access

Plik login.access także jest częścią pakietu sys-apps/shadow. Daje możliwość skonfigurowania tabeli dostępu do logowania. Dzięki tej tabeli można decydować czy ktoś będzie mógł się zalogować na podstawie swojej nazwy użytkownika, przynależności do grupy czy nazwy hosta. Domyślnie taką możliwość mają wszyscy użytkownicy w systemie, więc plik ten zawiera jedynie zakomentowane przykładowe wartości ustawień. Niezależnie od tego czy konfigurujemy serwer czy stację roboczą warto zablokować dostęp do konsoli wszystkim poza administratorami.

Uwaga: Te ustawienia nie dotyczą roota.

Listing 8: /etc/login.access

```
-:ALL EXCEPT wheel sync:console  
-:wheel:ALL EXCEPT LOCAL .gentoo.org
```

Ważne: Należy zachować ostrożność przy tym ustawieniu. Zły wpis skończy się utratą dostępu do komputera jeżeli nie masz dostępu do konta root.

Uwaga: Ustawienie te nie mają wpływu na SSH, ponieważ SSH nie używa domyślnie /bin/login. Możemy jednak analogicznie ustawić SSH, dodając linię UseLogin yes w /etc/ssh/sshd_config.

Takie ustawienie umożliwia zalogowanie się jedynie członkom grupy wheel z domeny gentoo.org. Może jest to dość paranoiczne podejście, lepiej jednak być trochę "nadmiernie" zabezpieczonym niż później żałować.

6. Prawa dostępu do plików

6.a. Dostępne do odczytu dla wszystkich

Zwykli użytkownicy nie powinni mieć dostępu do plików konfiguracyjnych i zawierających hasła. Jeśli włamywaczowi uda się wykrąść hasła z bazy danych lub strony internetowej to będzie mógł podmienić albo nawet usunąć nasze pliki. Właśnie to jest powodem, dla którego warto uważnie ustawić prawa do wszystkich ważnych plików. Jeśli jest się pewnym, że z danego pliku będzie korzystał wyłącznie root należy ustawić jego prawa na 0600 i przypisać plik odpowiedniemu użytkownikowi przy pomocy polecenia chown.

6.b. Dostępne do zapisu dla wszystkich

Listing 1: Znajdowanie takich plików i katalogów

```
# /usr/bin/find / -type f \(\ -perm -2 -o -perm -20 \) \
    -exec ls -lg {} \; 2>/dev/null >writable.txt
# /usr/bin/find / -type d \(\ -perm -2 -o -perm -20 \) \
    -exec ls -ldg {} \; 2>/dev/null >>writable.txt
```

Polecenie to utworzy listę plików z prawami do zapisu dla wszystkich użytkowników lub grup. Należy sprawdzić prawa, a następnie usunąć zapisywalność przy pomocy polecenia /bin/chmod o-w dla każdego z plików.

6.c. Pliki z bitami SUID i SGID

Pliki z ustawionymi bitami SUID i SGID uruchamiane są zawsze z prawami dostępu swojego właściciela, a nie z prawami osoby je uruchamiającej. Normalnie bity te są ustawiane na programach, które do poprawnej pracy wymagają praw roota. Jeśli taki plik zawiera błąd może spowodować kompromitację lokalnego konta root. W związku z takim niebezpieczeństwem należy za wszelką cenę unikać ustawiania tych bitów. Jeśli któryś z takich plików nie jest używany wykonujemy na nim polecenie chmod 0 lub odmiergowujemy pakiet do którego należy (pakiet do którego należą dane pliki wyszukujemy przy pomocy polecenia equery; jeśli nie ma takiego polecenia należy zainstalować pakiet gentoolkit). W każdym innym wypadku należy użyć polecenia chmod -s do zdjęcia bitu SUID.

Listing 2: Znajdowanie plików z bitem SUID

```
# /usr/bin/find / -type f \(\ -perm -004000 -o -perm -002000 \) \
    -exec ls -lg {} \; 2>/dev/null >suidfiles.txt
```

To polecenie utworzy plik z listą wszystkich programów, które mają ustawione bity SUID i SGID.

Listing 3: Lista plików binarnych z uidem

```
/bin/su
/bin/ping
/bin/mount
/bin/umount
/var/qmail/bin/qmail-queue
/usr/bin/chfn
/usr/bin/chsh
/usr/bin/crontab
/usr/bin/chage
/usr/bin/expiry
/usr/bin/perl5.6.1
/usr/bin/newgrp
/usr/bin/passwd
/usr/bin/gpasswd
/usr/bin/procmail
/usr/bin/suidperl
/usr/lib/misc/pt_chown
/usr/sbin/unix_chkpwd
/usr/sbin/traceroute
/usr/sbin/pwd_db_chkpwd
```

Domyślnie Gentoo nie zawiera wielu plików SUID (choć ich ilość zależy od tego jakie programy są zainstalowane). Większość

komend nie powinna być uruchamiana przez zwykłych użytkowników z prawami roota. Polecamy zdjęcie bitów SUID z programów ping, mount, umount, chfn, chsh, newgrp, suidperl, pt_chown i traceroute. Dokonuje się tego przy pomocy polecenia chmod -s. Nie należy usuwać tego bitu z su, qmail-queue czy unix_chkpwd. Usunięcie setuid z tych plików spowoduje utratę możliwości przełączania się na konto superużytkownika czy otrzymywania poczty. Usuwając SUID odbieramy zwykłemu użytkownikowi (lub włamywaczowi) możliwość uzyskania dostępu do konta roota przez któryś z tych plików.

Jedyne plikami z SUID w moim systemie to: su, passwd, gpasswd, qmail-queue, unix_chkpwd i pwdb_chkpwd. Każdy kto używa serwera X może potrzebować jeszcze kilku, ponieważ serwer korzysta czasem z rozszerzonego dostępu, oferowanego przez bity SUID.

6.d. Twarde dowiązania do plików binarnych z SUID/SGID

Plik można uznać za skasowany tylko wtedy gdy nie istnieją żadne na niego wskazujące odnośniki. Może to wydawać się dziwne, ale na przykład plik /usr/bin/perl jest wyłącznie odnośnikiem do inody gdzie znajdują się właściwe dane. Do pliku może istnieć nieskończonie duża ilość dowiązań i dopóki wszystkie nie zostaną skasowane plik wciąż istnieje.

Jeśli użytkownicy posiadają dostęp do partycji nie zamontowanej z opcjami nosuid i noexec (na przykład gdy /tmp, /home, lub /var/tmp nie znajdują się na osobnych partycjach) należy się upewnić, że nie będą mieli możliwości utworzenia trwałego dowiązania do plików binarnych i korzystania z nich np. po aktualnieniu przez Portage pliku do nowej, nie zawierającej błędów wersji.

Ostrzeżenie: Jeśli otrzymamy od Portage ostrzeżenie o istniejących twardych dowiązaniach, a użytkownicy mają dostęp do partycji umożliwiającej wykonywanie plików SUID/SGID należy uważnie przeczytać ten rozdział. Jeden z użytkowników może podejmować próbę obejścia uaktualnień poprzez zatrzymanie przestarzałej wersji programu. Jeśli użytkownicy nie mogą tworzyć własnych plików SUID lub uruchamiać programów poprzez dynamiczny program ładujący (partycje zamontowane jako noexec) nie ma powodu do zmartwień.

Uwaga: Użytkownik nie musi posiadać prawa odczytu pliku, aby móc utworzyć do niego dowiązanie, wystarczy, że ma prawa do katalogu go zawierającego.

Aby sprawdzić ilość istniejących dowiązań do danego pliku należy skorzystać z polecenia stat.

Listing 4: Polecenie stat

```
$ stat /bin/su
  File: `/bin/su'
  Size: 29350          Blocks: 64          IO Block: 131072 regular file
Device: 900h/2304d      Inode: 2057419      Links: 1
Access: (4711/-rws--x--x)  Uid: (     0/    root)   Gid: (     0/    root)
Access: 2005-02-07 01:59:35.000000000 +0000
Modify: 2004-11-04 01:46:17.000000000 +0000
Change: 2004-11-04 01:46:17.000000000 +0000
```

Do znajdowania plików z ustawnionymi bitami SUID i SGID korzystamy z programu find.

Listing 5: Znajdowanie wielokrotnie zlinkowanych plików binarnych z bitem uid/sgid

```
$ find / -type f \(\ -perm -004000 -o -perm -002000 \) -links +1 -ls
```

7. PAM

7.a. PAM

PAM jest zestawem bibliotek, które umożliwiają korzystanie z różnych metod autoryzacji w programach. Flaga USE pam jest włączana domyślnie. Domyślne ustawienia PAM w Gentoo są stosunkowo dobre, zawsze jednak można je nieco ulepszyć. Po pierwsze warto zainstalować sobie cracklib.

Listing 1: Instalowanie cracklib

```
# emerge cracklib
```

Listing 2: /etc/pam.d/passwd

```
auth      required pam_unix.so shadow nullok
account  required pam_unix.so
password required pam_cracklib.so difok=3 retry=3 minlen=8 dcredit=-2 ocredit=-2
password required pam_unix.so md5 use_authok
session   required pam_unix.so
```

Takie ustawienie zmusi użytkowników do ustawiania co najmniej ośmioliterowych haseł, zawierających co najmniej dwie cyfry i co najmniej dwa inne znaki oraz co najmniej trzy znaki różne w stosunku do tych z poprzedniego hasła. Spowoduje to, że wszyscy użytkownicy będą mieli ustawione dobre hasła. Więcej opcji znajduje się w dokumentacji [PAM](#).

Listing 3: /etc/pam.d/sshd

```
auth      required pam_unix.so nullok
auth      required pam_shells.so
auth      required pam_nologin.so
auth      required pam_env.so
account  required pam_unix.so
password required pam_cracklib.so difok=3 retry=3 minlen=8 dcredit=-2 ocredit=-2 use_authok
password required pam_unix.so shadow md5
session   required pam_unix.so
session   required pam_limits.so
```

Każda nieskonfigurowana w pliku /etc/pam.d usługa zostanie potraktowana zgodnie z zasadami z pliku /etc/pam.d/other. Domyślnie jest to deny, tak jak powinno to być zresztą ustawione. Osobiście lubię mieć dużą ilość logów, w związku z czym dodałem pam_warn.so. Ostatnim ustawieniem jest pam_limits, które jest konfigurowane w pliku /etc/security/limits.conf. Więcej informacji o tych ustawieniach znajduje się w paragrafie dotyczącym /etc/security/limits.conf.

Listing 4: /etc/pam.d/other

```
auth      required pam_deny.so
auth      required pam_warn.so
account  required pam_deny.so
account  required pam_warn.so
password required pam_deny.so
password required pam_warn.so
session   required pam_deny.so
session   required pam_warn.so
```

8. TCP Wrappers

8.a. TCP Wrappers

Jest to metoda kontrolowania dostępu do usług normalnie uruchamianych przez inetd (którego nie ma Gentoo), ale może również być używana przez xinetd i inne tego typu usługi.

Uwaga: Usługa ta powinna być uruchamiana z argumentem serwera tcpd (w xinetd). Więcej informacji znajduje się w paragrafie dotyczącym xinetd.

Listing 1: /etc/hosts.deny

```
ALL:PARANOID
```

Listing 2: /etc/hosts.allow

```
ALL: LOCAL @wheel  
time: LOCAL, .gentoo.org
```

Jak widać format tego pliku jest bardzo podobny do tego z pliku /etc/login.access. Tcpd posiada wsparcie dla określonych usług i nie nakłada się z /etc/login.access. Te ustawienie będą miały wpływ wyłącznie na usługi używające tcp wrappers.

Możliwe jest również wykonywanie poleceń podczas dostępu do usługi (np. jeśli chcemy włączyć przekazywanie dla użytkowników korzystających z łączności komutowanej) ale nie jest to zalecane, ponieważ ludzie mają tendencję do komplikowania prostych spraw. Przykładem takiej sytuacji może być przypadek, gdy chcemy skonfigurować skrypt, który będzie wysyłał e-mail za każdym razem gdy ktoś zostanie zakwalifikowany do reguły deny. Włamywacz może wykorzystać taką regułę do ataku typu DoS poprzez jej nadużywanie, co skończy się ogromnym obciążeniem systemu i wielką ilością e-maili w naszej skrzynce. Dlatego nie należy tego robić. Więcej informacji znajduje się w dokumentacji man 5 hosts_access.

9. Bezpieczeństwo jądra.

9.a. Usuwanie zbędnych funkcji

Podstawową zasadą przy konfigurowaniu jądra jest to, by wyłączać wszystko czego się nie potrzebuje. Dzięki temu stworzymy nie tylko małe, ale i mniej podatne na potencjalne błędy jądro.

Ponadto warto rozważyć wyłączenie opcji 'loadable module support' - wsparcia dla ładowania modułów. Nie uniemożliwi to zainstalowania rootkitu przez moduły jądra, ale na pewno znacznie to utrudni - zwłaszcza mniej wyedukowanym włamywaczom.

9.b. System plików proc

Wiele parametrów jądra można skonfigurować poprzez system plików /proc lub przy pomocy sysctl.

Możliwość zmiany tych parametrów bez konieczności restartowania komputera umożliwia zaznaczenie opcji CONFIG_SYSCTL w konfiguracji jądra. Jest ona standardowo zaznaczona w kernelach serii 2.4.

Listing 1: Wylaczanie przekazywania IP

```
# /bin/echo "0" > /proc/sys/net/ipv4/ip_forward
```

Należy upewnić się, że przekazywanie (forwardowanie) IP jest wyłączone. Jest ono potrzebne jedynie w przypadku stacji roboczych czy węzłów sieciowych o wielu połączeniach. Radzimy wyłączyć tą flagę na samym początku, ponieważ zmienia ona wartość kilku innych omawianych tu opcji.

Listing 2: Upuszczanie pakietów ping

```
# /bin/echo "1" > /proc/sys/net/ipv4/icmp_echo_ignore_all
```

Włączenie tej opcji spowoduje, że kernel po prostu zignoruje wszystkie wiadomości typu ping (czyli wiadomości typu 0 protokołu ICMP). Uzasadnieniem takiej decyzji jest to, że pakiet IP przenoszący wiadomość ICMP może przenosić ładunek inny niż możliwa by się po nim spodziewać. Administratorzy często używają ping jako narzędzia diagnostycznego i narzekają jeśli jest wyłączony, ale są również powody dla których warto zabronić pingowania serwera ludziom z zewnątrz. Czasem zdarza się sytuacja, że ktoś z wewnętrz będzie chciał użyć ping, dlatego znacznie lepszym pomysłem jest wyłączenie tego typu pakietów w firewallu i umożliwienie korzystania z tego narzędzia administratorom naszej sieci.

Listing 3: Ignorowanie transmisji ping

```
# /bin/echo "1" > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
```

Opcja ta wyłączy odpowiedzi na rozgłaszenie ICMP (ICMP broadcast) co zapobiegnie atakom typu smurf. Atak typu smurf polega na wysłaniu wiadomości ICMP typu 0 (czyli ping) na adres całej sieci ze sfałszowanego adresu, co spowoduje odpowiedź wszystkich działających w tej sieci komputerów i zalania falą odpowiedzi komputera, którego adresu użył atakujący.

Listing 4: Wylaczanie source routingu

```
# /bin/echo "0" > /proc/sys/net/ipv4/conf/all/accept_source_route
```

Ustawienie to spowoduje odrzucanie source routowanych pakietów. Włamywacze mogą ich używać do stworzenia pozorów, że ruch pochodzący od nich tak naprawdę pochodzi z wnętrza sieci, a faktycznie będzie routowany z powrotem do ich komputera. Source routingu bardzo rzadko używa się do legalnych czynności, więc można go bezpiecznie wyłączyć.

Listing 5: Wylaczanie akceptowania przedresowań

```
# /bin/echo "0" > /proc/sys/net/ipv4/conf/all/accept_redirects
# /bin/echo "0" > /proc/sys/net/ipv4/conf/all/secure_redirects
```

Spowoduje to odrzucanie przeadresowanych pakietów ICMP, które mogą być używane do zmiany tablic routingu, co zwykle źle się kończy.

Listing 6: Ochrona przed błędymi informacjami o błędach

```
# /bin/echo "1" > /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses
```

Włącza ochronę przed błędymi komunikatami o błędach.

Listing 7: Włączanie filtrowania trasy powrotnej (reverse path filtering)

```
# for i in /proc/sys/net/ipv4/conf/*; do
    /bin/echo "1" > $i/rp_filter
done
```

Włącza filtrowanie trasy powrotnej pakietu. Dzięki temu wszystkie pakiety, które nie docierają interfejsem, z którego rzekomo wyszło zapytanie zostaną zignorowane. Znacznie poprawia to bezpieczeństwo, ponieważ utrudnia spoofowanie IP. Aby uzyskać pełną funkcjonalność musimy je uruchomić dla każdego net/ipv4/conf/*.

Listing 8: Logowanie wszystkich spoofowanych, source routowanych lub przeadresowanych pakietów

```
# /bin/echo "1" > /proc/sys/net/ipv4/conf/all/log_martians
```

Loguje wszystkie spoofowane, source routowane i przeadresowane pakiety.

Wszystkie wprowadzone w powyższy sposób zmiany zostaną zresetowane po ponownym uruchomieniu komputera. Aby zapamiętać je na dłużej należy dopisać je do pliku /etc/sysctl.conf, z którego informacje są automatycznie wczytywane przez skrypt startowy /etc/init.d/bootmisc.

Składnia pliku /etc/sysctl.conf jest bardzo przejrzysta. Wystarczy usunąć z podanych wyżej linii /proc/sys/ i zamienić wszystkie / na ..

Listing 9: Tłumaczenie na składnię sysctl.conf

```
(Ręcznie, przy użyciu echo):
/bin/echo "0" > /proc/sys/net/ipv4/ip_forward

(Automatycznie w sysctl.conf:)
net.ipv4.ip_forward = 0
```

9.c. Grsecurity

Łatka grsecurity jest standardowo nakładana na jądra sys-kernel/hardened-sources w Gentoo, chociaż w domyślnym pliku konfiguracyjnym wyłączamy wszystkie dostarczane przez nią moduły. Najpierw konfiguruje się standardowe opcje kernela, potem przychodzi pora na dodanie tych związanych z grsecurity. Dokładne omówienie wszystkich użytych ustawień znajduje się na stronie projektu Gentoo Hardened.

Najnowsze hardened-sources są oparte na wersjach 2.* Grsecurity. Więcej informacji o tych łatkach znajduje się na stronach projektu Grsecurity.

9.d. Kerneli

Kerneli jest to łatka dodająca do jądra funkcje związane z szyfrowaniem takie jak szyfry kryptograficzne, algorytmy skrótów czy filtry pętli kryptograficznych.

Ostrzeżenie: Przy używaniu kerneli trzeba zachować ostrożność, nie jest to jeszcze wersja stabilna.

10. Zabezpieczanie usług

10.a. Apache

Apache (1.3.26) posiada stosunkowo dobry domyślny plik konfiguracyjny, warto jednak nanieść na niego kilka poprawek, jak na przykład związanie go z jednym adresem czy zapobieganie wyciekom informacji. Poniżej podajemy opcje, które warto dodać do pliku konfiguracyjnego.

O ile nie wyłączono ssl w pliku /etc/make.conf przed instalacją, Apache powinien posiadać jego obsługę. Aby ją uaktywnić należy dopisać następującą linię:

Listing 1: /etc/conf.d/apache

```
HTTPD_OPTS="-D SSL"
```

Listing 2: /etc/apache/conf/apache.conf

```
#Make it listen on your ip
Listen 127.0.0.1
BindAddress 127.0.0.1
#It is not a good idea to use nobody or nogroup -
#for every service not running as root
#(just add the user apache with group apache)
User apache
Group apache
#Will keep apache from telling about the version
ServerSignature Off
ServerTokens Prod
```

Apache jest komplikowany z opcjami --enable-shared=max i --enable-module=all, czyli ze wszystkimi modułami, co zmusza do wykomentowywania wszystkich przez nas nie używanych z sekcji LoadModule (LoadModule i AddModule). Następnie należy zrestartować usługę polecienniem /etc/init.d/apache restart.

Całość dokumentacji znajduje się na stronie <http://www.apache.org>.

10.b. Bind

Dokumentacja znajduje się na stronach [Internet Software Consortium](http://www.isc.org). "The BIND 9 Administrator Reference Manual" znajduje się również w katalogu doc/arm.

Nowsze ebuildy BIND umożliwiają chrootowanie się poza maszynę. Po zemergowaniu bind należy wykonać następujące czynności:

Listing 3: Chrootowanie BIND

```
ebuild /var/db/pkg/net-dns/bind-9.2.2-r2/bind-9.2.2-r2.ebuild config\`"
(Przed wpisaniem powyższego polecenia warto zmienić katalog dla chroota
w pliku /etc/conf.d/named. W innym wypadku zostanie użyty katalog /chroot/dns)
(Należy zamienić podany numer wersji aktualnie używanym)
```

10.c. Djbdns

Djbdns to implementacja zabezpieczeń DNS, której autor jest tak pewny, że stawia na nią [pieniądze](#). Jej sposób działania bardzo różni się od tego z Bind 9, ale mimo wszystko warto ją wypróbować.Więcej informacji znajduje się na stronie <http://www.djbdns.org>.

10.d. FTP

Generalnie używanie FTP (File Transfer Protocol) to kiepski pomysł. FTP używa niezaszyfrowanych haseł (są przesyłane czystym tekstem), nasłuchuje na dwóch portach (standardowo 20 i 21), a ponadto jest często atakowane przez włamywaczy szukających anonimowych serwerów w celu wymiany warezów. W związku z wieloma brakami w bezpieczeństwie protokołu FTP powinno się zamiast niego używać sftp lub HTTP. Jeśli nie jest to możliwe należy zabezpieczyć FTP jak to tylko możliwe i przygotować się na sporo problemów.

10.e. Mysql

Jeśli dostęp do bazy mysql jest potrzebny wyłącznie lokalnym aplikacjom należy odkomentować następującą linię w pliku /etc/mysql/my.cnf:

Listing 4: Zamknięcie dostępu z sieci

```
skip-networking
```

Następnie wyłączymy komendę LOAD DATA LOCAL INFILE. Zapobiegnie to nieautoryzowanemu dostępowi do lokalnych plików. Takie zabezpieczenie przydaje się np. w przypadku odkrycia słabych punktów w PHP umożliwiających atak typu SQL Injection.

Listing 5: Wylaczanie LOAD DATA LOCAL INFILE w sekcji [mysqld]

```
set-variable=local-infile=0
```

Następnie musimy usunąć przykładową bazę danych (test) i wszystkie konta poza kontem roota.

Listing 6: Usuwanie przykładowej bazy i zbędnych kont

```
mysql> drop database test;
mysql> use mysql;
mysql> delete from db;
mysql> delete from user where not (host="localhost" and user="root");
mysql> flush privileges;
```

Ostrzeżenie: Należy uważać z wpisywaniem powyższych poleceń jeśli posiada się już skonfigurowane konta użytkowników.

Uwaga: Jeśli zmieniano hasła z wiersza poleceń MySQL należy wyczyścić pliki `~/.mysql_history` i `/var/log/mysql/mysql.log` ponieważ zawierają one historię poleceń SQL wraz z hasłami zapisanymi otwartym tekstem.

10.f. Proftpd

Proftpd miało wiele problemów związanych z bezpieczeństwem, na szczęście większość z nich już naprawiono. Do domyślnego pliku konfiguracyjnego warto dodać następujące opcje:

Listing 7: /etc/proftpd/proftpd.conf

```
ServerName "My ftp daemon"
#Don't show the ident of the server
ServerIdent on "Go away"

#Makes it easier to create virtual users
RequireValidShell off

#Use alternative password and group file (passwd uses crypt format)
AuthUserFile "/etc/proftpd/passwd"
AuthGroupFile "/etc/proftpd/group"

# Permissions
Umask 077

# Timeouts and limitations
MaxInstances 30
MaxClients 10 "Only 10 connections allowed"
MaxClientsPerHost 1 "You have already logged on once"
MaxClientsPerUser 1 "You have already logged on once"
TimeoutStalled 10
TimeoutNoTransfer 20
TimeoutLogin 20

#Chroot everyone
DefaultRoot ~
```

```
#don't run as root
User nobody
Group nogroup

#Log every transfer
TransferLog /var/log/transferlog

#Problems with globbing
DenyFilter \*.*/
```

Całość dokumentacji znajduje się na stronie <http://www.proftpd.org>.

10.g. Pure-ftp

Pure-ftp to odmiana trollftpd, zmodyfikowana pod kątem bezpieczeństwa przez Franka Dennisa.

Dzięki włączeniu opcji AUTH uniemożliwia się korzystania z kont użytkowników systemowych i zmusza serwer do pracy z kontami wirtualnymi. Bazę danych wirtualnych użytkowników tworzy się poleceniem -lpuredb:/etc/pureftpd.pdb, a użytkowników do niej dodaje za pomocą /usr/bin/pure-pw.

Listing 8: /etc/conf.d/pure-ftpd

```
AUTH="-lpuredb:/etc/pureftpd.pdb"

## Misc. Others ##
MISC_OTHER="-A -E -X -U 177:077 -d -4 -L100:5 -I 15"
```

Warto również skonfigurować opcję MISC_OTHER, tak aby uniemożliwić logowanie się jako anonymous (-E), zamknąć każdego użytkownika w chroocie (-A), uniemożliwić dostęp do plików rozpoczynających się od . (kropki) (-X), podać maksymalny czas beczynności (-I), zabronić rekursywnego pobierania (-L) oraz ustawić sensowny umask.

Ostrzeżenie: Nie wolno używać opcji -w lub -W! Jeśli chcesz założyć warez zrezygnuj z czytania tego przewodnika!

Całość dokumentacji znajduje się na stronie <http://www.pureftpd.org>.

10.h. Vsftpd

Vsftpd (skrót od "bardzo bezpieczne ftp" (very secure ftp)) to mały daemon ftp z bardzo dobrą domyślną konfiguracją. Jest bardzo prostym programem, więc nie posiada tak rozbudowanych możliwości jak pureftpd albo proftpd.

Listing 9: /etc/vsftpd

```
anonymous_enable=NO
local_enable=YES

#read only
write_enable=NO

#enable logging of transfers
xferlog_std_format=YES

idle_session_timeout=20
data_connection_timeout=20
nopriv_user=nobody

chroot_list_enable=YES
chroot_list_file=/etc/vsftpd/chrootlist

ls_recurse_enable=NO
```

Jak widać nie ma możliwości, aby usługa ta posiadała jakieś indywidualne ustawienia praw dostępu. Jeśli jednak chodzi o ustawienia dla anonimowych użytkowników okazuje się ona bardzo dobra, a taki anonimowy serwer przydaje się np. gdy trzeba szybko podzielić się z kimś kodem open source.

10.i. Qmail

Qmail jest uważany za bardzo bezpieczny serwer pocztowy. Przy jego pisaniu zwrócono szczególną uwagę na opcje związane z bezpieczeństwem. Domyślana konfiguracja nie umożliwia przekazywania poczty, a w samym programie nie znaleziono błędu związanego z bezpieczeństwem od 1996 roku. Nie ma się co zastanawiać, należy po prostu wpisać emerge qmail i zabrać się za konfigurację.

10.j. Samba

Samba jest protokołem umożliwiającym wymianę plików z sieciami Microsoft i Novell, z którego nie powinno korzystać się w Internecie. Ponadto jak niemal każdy program wymaga pewnego dokonfigurowania.

Listing 10: /etc/samba/smb.conf

```
[global]
#Bind to an interface
interfaces = eth0 10.0.0.1/32

#Make sure to use encrypted password
encrypt passwords = yes
directory security mask = 0700

#allow traffic from 10.0.0./*
hosts allow = 10.0.0.

#Enables user authentication
#(don't use the share mode)
security = user

#Disallow privileged accounts
invalid users = root @wheel

#Maximum size smb shows for a share (not a limit)
max disk size = 102400

#Uphold the password policy
min password length = 8
null passwords = no

#Use PAM (if added support)
obey pam restrictions = yes
pam password change = yes
```

Warto upewnić się, że wszystkie prawa dostępu są ustawione prawidłowo oraz zapoznać się z [dokumentacją](#) programu.

Następnie należy zrestartować serwer i dodać konta użytkowników, którzy mają mieć dostęp do tej usługi. Dokonuje się tego przy pomocy polecenia /usr/bin/smbpasswd z parametrem -a.

10.k. ssh

Jedynym zabezpieczeniem jakiego potrzebuje OpenSSH jest przełączenie go na silniejszy mechanizm uwierzytelniania, oparty na szyfrowaniu klucza publicznego (przy pomocy niewłaściwie ustawionych haseł włamano się już między innymi na strony <http://www.sourceforge.net>, <http://www.php.net> i <http://www.apache.org>).

Listing 11: /etc/ssh/sshd_config

```
#Only enable version 2
Protocol 2

#Disable root login. Users have to su to root
PermitRootLogin no

#Turn on Public key authentication
```

```

PubkeyAuthentication yes
AuthorizedKeysFile      .ssh/authorized_keys

#Disable .rhost and normal password authentication
RhostsAuthentication no
PasswordAuthentication no
PermitEmptyPasswords no

#Only allow userin the wheel or admin group to login
AllowGroups wheel admin

#In those groups only allow the following users
#The @<domainname> is optional but replaces the
#older AllowHosts directive
AllowUsers kn@gentoo.org bs@gentoo.org

#Logging
SyslogFacility AUTH
LogLevel INFO

ListenAddress 127.0.0.1

```

Należy sprawdzić czy w pliku konfiguracyjnym nie ma linii UsePAM yes, ponieważ nadpisze ona ustawienia dla uwierzytelniania poprzez klucze publiczne.

Teraz wszyscy użytkownicy muszą wygenerować własne klucze na komputerach, z których będą się logować na nasz serwer. Dokonuje się tego następującym poleceniem:

Listing 12: Tworzenie pary kluczy DSA

```
# /usr/bin/ssh-keygen -t dsa
```

Następnie należy wpisać hasło.

Listing 13: Wynik polecenia ssh-keygen

```

Generating public/private dsa key pair.
Enter file in which to save the key (/home/kn/.ssh/id_dsa):[Press enter]
Created directory '/home/kn/.ssh'.
Enter passphrase (empty for no passphrase): [Enter passphrase]
Enter same passphrase again: [Enter passphrase again]
Your identification has been saved in /home/kn/.ssh/id_dsa.
Your public key has been saved in /home/kn/.ssh/id_dsa.pub.
The key fingerprint is:
07:24:a9:12:7f:83:7e:af:b8:1f:89:a3:48:29:e2:a4 kn@knielsen

```

W wyniku tych czynności w katalogu `~/.ssh/` zostaną utworzone dwa pliki o nazwach `id_dsa` i `id_dsa.pub`. Plik o nazwie `id_dsa` to prywatny klucz i nie powinien być przekazywany innym osobom. Natomiast drugi plik, `id_dsa.pub`, powinno umieścić się w `~/.ssh/authorized_keys` na każdym serwerze, na który zamierzamy się logować:

Listing 14: Dodawanie pliku id_dsa.pub do pliku authorized_keys

```

$ scp id_dsa.pub other-host:/var/tmp/currenthostname.pub
$ ssh other-host
password:
$ cat /var/tmp/currenthostname.pub >> ~/.ssh/authorized_keys

```

Użytkownicy powinni pilnować swoich prywatnych kluczy i przechowywać je wyłącznie na nośnikach, które będą zawsze mieć przy sobie lub na swoich stacjach roboczych. Informację o tym należy umieścić w [polityce bezpieczeństwa](#) dotyczącej haseł.

10.1. Używanie xinetd

xinetd to zastępstwo dla demona usług internetowych - inetd (którego nie ma w Gentoo). Umożliwia kontrolę dostępu na podstawie adresu i czasu tego dostępu. Posiada również szerokie możliwości logowania, obejmujące między innymi czas uruchomienia serwera, zdalne adresy hostów, zdalnych użytkowników, czas działania serwera i wszystkie zgłoszone żądania.

Jak w przypadku wszystkich innych usług ważną rzeczą jest tu bardzo uważna konfiguracja. Ponieważ jednak xinetd jest uruchamiane z konta roota i obsługuje wiele protokołów, których działanie nie jest powszechnie znane, odradzamy jego używanie. Jeśli jednak koniecznie chcecie go używać oto jak należy go zabezpieczyć:

Listing 15: Instalowanie xinetd

```
# emerge xinetd tcp-wrappers
```

Następnie edytujemy plik konfiguracyjny:

Listing 16: /etc/xinetd.conf

```
defaults
{
    only_from = localhost
    instances = 10
    log_type = SYSLOG authpriv info
    log_on_success = HOST PID
    log_on_failure = HOST
    cps = 25 30
}

# This will setup pserver (cvs) via xinetd with the following settings:
# max 10 instances (10 connections at a time)
# limit the pserver to tcp only
# use the user cvs to run this service
# bind the interfaces to only 1 ip
# allow access from 10.0.0.-
# limit the time developers can use cvs from 8am to 5pm
# use tpcd wrappers (access control controlled in
# /etc/hosts.allow and /etc/hosts.deny)
# max_load on the machine set to 1.0
# The disable flag is per default set to no but I like having
# it in case of it should be disabled
service cvspserver
{
    socket_type = stream
    protocol = tcp
    instances = 10
    protocol = tcp
    wait = no
    user = cvs
    bind = 10.0.0.2
    only_from = 10.0.0.0
    access_times = 8:00-17:00
    server = /usr/sbin/tcpd
    server_args = /usr/bin/cvs --allow-root=/mnt/cvssdisk/cvsroot pserver
    max_load = 1.0
    log_on_failure += RECORD
    disable = no
}
```

Więcej informacji znajduje się w man 5 xinetd.conf.

10.m. X

Domyślnie Xorg jest skonfigurowany jako serwer X. Może to być niebezpieczne, ponieważ X używa nieszyfrowanych połączeń TCP i oczekuje na połączenia od xklientów.

Ważne: Jeśli ta usługa nie jest niezbędna należy ją po prostu wyłączyć.

Jeśli jednak komputer ma działać jako serwer X należy używać polecenia /usr/X11R6/bin/xhost z ostrożnością. Polecenie to umożliwia łączenie się z naszym serwerem klientom z innych hostów i używanie naszego ekranu. Może to być pomocne, jeśli chcemy użyć aplikacji opartej na X na innym komputerze, do którego dostęp mamy jedynie przez sieć, ale może również zostać wykorzystane w nieczytliwych celach przez włamywaczy. Składnia dla tej komendy to /usr/X11R6/bin/xhost +hostname.

Ostrzeżenie: Nie wolno używać składni xhost +! Umożliwi to połączenie z serwerem i przejęcie nad nim kontroli dowolnemu klientowi. Jeśli włamywacz uzyska dostęp do X może bez trudu logować sekwencje klawiszy wpisywane przez użytkowników, poznać ich hasła i w efekcie tego przejąć całkowitą kontrolę nad maszyną. Jeśli używanie X jako serwera jest konieczne należy zawsze wyszczególnić prawidłowe hosty.

Dużo bezpieczniejszym rozwiązaniem jest nie korzystanie z tej możliwości np. poprzez uruchamianie serwera X poleceniem startx --nolisten tcp lub całkowite jej wyłączenie w plikach konfiguracyjnych.

Listing 17: /usr/X11R6/bin/startx

```
defaultserverargs="-nolisten tcp"
```

Aby uniemożliwić nadpisanie pliku startx w trakcie emergowania nowych wersji Xorg należy dodać go do listy plików chronionych, dokonując się tego poprzez dopisanie następującej linii do pliku /etc/make.conf:

Listing 18: /etc/make.conf

```
CONFIG_PROTECT_MASK="/usr/X11R6/bin/startx"
```

Graficzny menedżer logowania wymaga nieco innego podejścia:

Dla gdm (Gnome Display Manager):

Listing 19: /etc/X11/gdm/gdm.conf

```
[server-Standard]
command=/usr/X11R6/bin/X -nolisten tcp
```

Dla xdm (X Display Manager) i kdm (Kde Display Manager):

Listing 20: /etc/X11/xdm/Xservers

```
:0 local /usr/bin/X11/X -nolisten tcp
```

11. Chrootowanie i serwery wirtualne

11.a. Chrootowanie

Zamknięcie usługi (użytkownika) w chroocie uniemożliwi dostęp do plików i informacji, do których dostępu mieć nie powinna, a także zamknie wiele z możliwości uzyskania dostępu do konta roota tą drogą. Ponadto uruchomienie usługi z konta użytkownika innego niż root (np. nobody, apache, named) ograniczy włamywaczom dostęp, pozostawiając im do dyspozycji tylko pliki przypisane do tych kont. Takie postępowanie uniemożliwi także nieautoryzowany dostęp do konta w przypadku błędu związanego z bezpieczeństwem w którejś z usług.

Niektóre usługi, takie jak np. pure-ftpd i bind posiadają wbudowane możliwości zamykania się w chroocie. Jeśli jednak nie ma takiej możliwości należy wykonać to samodzielnie. Oto prosty przykład tworzenia środowiska chroot. Dla uproszczenia przetestujemy je przy pomocy basha.

Na początek tworzymy katalog /chroot przy pomocy polecenia mkdir /chroot oraz wyszukujemy wszystkie biblioteki, z którymi bash jest dynamicznie zlinkowany (jeśli skompilowano go z opcją -static można pominąć ten krok).

Listę bibliotek używanych przez bash uzyskamy dzięki poleceniu:

Listing 1: Uzyskiwanie listy używanych bibliotek

```
# ldd /bin/bash
libncurses.so.5 => /lib/libncurses.so.5 (0x4001b000)
libdl.so.2 => /lib/libdl.so.2 (0x40060000)
libc.so.6 => /lib/libc.so.6 (0x40063000)
/lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
```

Następnie tworzymy środowisko dla basha.

Listing 2: Tworzenie środowiska chroot dla basha

```
# mkdir /chroot/bash
# mkdir /chroot/bash/bin
# mkdir /chroot/bash/lib
```

Później kopujemy pliki używane przez bash (/lib) do katalogu lib w chroocie oraz plik binarny bash do nowego bin. Spowoduje to utworzenie kopii naszego środowiska, posiadającej jednak nieco mniej możliwości. Po utworzeniu warto je wypróbować poleceniem chroot /chroot/bash /bin/bash. Jeśli pokaże się znak zachęty w postaci / to znaczy, że wszystko działa prawidłowo. W przeciwnym wypadku zostaniemy poinformowani o braku pewnych plików. Niektóre współdzielone biblioteki są wzajemnie od siebie zależne.

Łatwo zauważyc, że wewnętrz chroota nie działa nic poza echo. Spowodowane jest to tym, że w naszym nowym środowisku jedynym dostępnym poleceniem jest bash, a echo jest jego wbudowaną funkcją.

Zamykanie usług w chroocie wygląda bardzo podobnie. Jedyną różnicą jest to, że usługom potrzebne są czasem urządzenia i pliki konfiguracyjne z /etc. Należy je więc po prostu skopiować do nowego środowiska (urządzenia kopiuje się poleceniem cp -a), należy również odpowiednio wyedytować skrypt startowy zanim skorzysta się z nowego środowiska. Czasem ciężko znaleźć wszystkie urządzenia i pliki jakich wymaga dana usługa, wtedy bardzo przydatny jest program strace. Uruchamiamy usługę poprzez /usr/bin/strace i szukamy wszystkich odwołań open, read, stat i connect. Zwykle naprowadza to na ślad wszystkich plików i urządzeń, które należy skopiować. W większości przypadków wystarczy skopiowanie pliku passwd (należy z niego usunąć wszystkie linie dotyczące użytkowników nie mających nic wspólnego z usługą), oraz /dev/zero, /dev/log i /dev/random.

11.b. User Mode Linux

Kolejnym dobrym sposobem zabezpieczenia środowiska pracy jest uruchomienie wirtualnej maszyny. Taka maszyna to proces działający jako nakładka na właściwy system operacyjny, na który składa się drugi OS i pewna konfiguracja urządzeń, tworząca złudzenie, że jest właściwym komputerem. Korzyści z punktu widzenia zabezpieczeń systemu są takie, że w przypadku włamania zostanie pokonana tylko wirtualna maszyna, a właściwy system operacyjny pozostanie nienaruszony.

Więcej informacji o konfigurowaniu User Mode Linux znajduje się w [User Mode Linux Guide](#).

12. Firewall

12.a. Firewall

Ludzie często sądzą, że firewall zapewnia pełną ochronę. Są w błędzie. W większości przypadków źle skonfigurowany firewall zapewnia mniej ochrony niż nie posiadanie firewalla w ogóle. Ponadto każdy firewall należy traktować jak zwykły, podatny na błędy program.

Należy zatem poważnie przemyśleć jego używanie i dokładnie sprawdzić czy jest nam potrzebny. Należy również spisać dokładnie jakiego typu ma to być firewall, określić reguły jego działania oraz użytkowników jacy będą z niego korzystać.

Firewall jest przydatny z dwóch powodów:

- Nie wpuszcza części niepożądanych użytkowników (robaków/włamywaczy)
- Wpuszcza całą resztę (pracowników/dzieci)

Są trzy typy firewalli:

- Zajmujące się filtrowaniem pakietów.
- Kontrolujące połączenia sieciowe między systemami wewnętrznyimi i zewnętrznyimi (circuit relay)
- Bramy aplikacyjne

Firewallem powinien być dedykowany komputer, na którym nie są uruchomione żadne usługi (oprócz sshd, które może stanowić tu wyjątek) i który jest zabezpieczony w sposób opisany w tym przewodniku.

12.b. Filtrowanie pakietów

Cały ruch sieciowy odbywa się za pomocą pakietów. Duże ilości danych są dzielone na małe pakiety, przesyłane, a następnie składane w całość na komputerze docelowym. W nagłówku każdego pakietu znajduje się informacja o tym w jaki sposób i dokąd powinien on zostać dostarczony. Właśnie z tych informacji korzysta filtr pakietów. Filtrowanie polega na:

- Dopuszczaniu lub odrzucaniu pakietów na podstawie ich źródłowego i docelowego adresu IP.
- Dopuszczaniu lub odrzucaniu pakietów na podstawie ich źródłowego i docelowego portu.
- Odrzucanie lub dopuszczanie na podstawie używanego protokołu.
- Odrzucaniu lub dopuszczaniu pakietów na podstawie ustawionych w nagłówku pakietu flag danego protokołu.

Innymi słowy filtrowanie pakietów opiera się na danych pochodzących z nagłówka pakietu, a nie danych jakie on przenosi.

Słabości:

- Informacje o adresie wewnętrz pakietów mogą zostać sfałszowane (zespoofowane) przez nadawcę.
- Treść dopuszczonego pakietu może zawierać niechciane informacje, których włamywacz może użyć do złamania aplikacji znajdujących się za firewallem.
- Może być zawodny

Mocne strony:

- Łatwość implementacji.
- Możliwość informowania o potencjalnych atakach zanim jeszcze się odbędą (np. poprzez wykrywanie skanowania portów).
- Doskonale nadaje się do zatrzymywania ataków typu SYN.

Przykładowe filtry pakietów na Linuksa:

- [Iptables](#)
- [Ipchains](#)
- [SmoothWall](#)

Uwaga: Zaleca się używanie iptables. Ipchains to przeżytek.

12.c. Circuit relay

Jest to firewall, który zatwierdza połączenia zanim pozwoli na wymianę jakichś danych. Oznacza to, że nie odrzuca i nie dopuszcza pakietów jedynie na podstawie ich nagłówków, ale na podstawie skonfigurowanego zestawu reguł sprawdza czy połączenie jest prawidłowe i dopiero kiedy to potwierdzi pozwala na wymianę informacji. Filtrowanie może się odbywać poprzez kontrolę:

- Źródłowego i docelowego adresu IP
- Źródłowego i docelowego portu
- Okresu czasu
- Protokołu
- Użytkownika
- Hasła

Cały ruch jest zatwierdzany i monitorowany, ten niechciany jest po prostu odrzucany.

Słabości:

- Działa na warstwie transportowej, przez co niektóre aplikacje mogą wymagać pewnych modyfikacji, aby mogły w normalny sposób spełniać swoje funkcje.

12.d. Bramy aplikacyjne (application gateway)

Brama aplikacyjna to rodzaj proxy dla programów, które wymienia dane klientów ze zdalnymi hostami w ich imieniu. Zwykle znajduje się poza widokiem publicznym i poza strefą zdemilitaryzowaną (de-militarized zone, część prywatnej sieci, która jest widoczna poprzez firewall) lub firewalllem odrzucającym wszystkie połączenia. Filtrowanie odbywa się na zasadzie:

- Odrzucania lub akceptowania na podstawie źródłowego i docelowego adresu IP.
- Zawartości pakietu.
- Limitowanym dostępem do plików opartym na ich typie i rozszerzeniu.

Korzyści:

- Pozwala na buforowanie (cache) plików, co zwiększa wydajność sieci.
- Dokładnie loguje wszystkie połączenia.
- Dobrze się skaluje (niektóre proxy mogą współdzielić buforowane dane)
- Uniemożliwia bezpośredni dostęp z zewnątrz.
- Może "w locie" zmieniać zawartość pakietu.

Wady:

- Skomplikowany proces konfiguracji.

Bramy aplikacyjne to najlepszy sposób zabezpieczania sieci, ponieważ nie muszą być uruchamiane z przywilejami roota oraz uniemożliwiają bezpośredni dostęp z Internetu do komputerów w sieci.

Przykład darmowej bramy aplikacyjnej:

- [Squid](#)

12.e. Iptables

Korzystanie z iptables jest możliwe tylko wtedy gdy włączy się ich obsługę w jądrze. Zwykle całość kompiluje się jako moduły, a polecenie iptables w razie potrzeby ładuje te z nich, które są mu potrzebne. Jeśli jednak zgodnie z poprzednimi sugestiami zrezygnowano z włączania opcji Loadable Kernel Modules można spokojnie wkompilować wszystko na stałe. Więcej informacji na temat tego jak skonfigurować jądro przed użyciem iptables znajduje się w dokumentacji [Iptables Tutorial Chapter 5: Preparations](#). Po skompilowaniu jądra z odpowiednimi opcjami (lub jeszcze w trakcie jego kompilowania) należy zainstalować program iptables przy pomocy polecenia emerge iptables.

Test czy program działa prawidłowo wykonuje się wpisując polecenie iptables -L. Jeśli się nie uda to znaczy, że coś poszło nie tak i należy sprawdzić swoją konfigurację.

Iptables to filtr pakietów, który wprowadzono do jądra Linuksa w serii 2.4.x. Jednym z głównych usprawnień w stosunku do poprzedniej wersji (ipchains), było dodanie dynamicznego filtrowania pakietów (stateful packet filtering). Dzięki dynamicznemu filtrowaniu pakietów możliwe jest śledzenie każdego nawiązanego połączenia TCP.

Każde połączenie TCP składa się z serii pakietów zawierających informacje o źródłowym i docelowym adresie IP, źródłowym i docelowym porcie oraz numerach porządkowych, tak aby mogły być bez utraty danych sortowane po odebraniu. TCP to protokół połączeniowy, w przeciwieństwie do np. protokołu UDP.

Dzięki analizie nagłówka pakietu TCP dynamiczny filtr pakietów jest w stanie ustalić czy otrzymany pakiet TCP jest częścią już nawiązanego połączenia oraz zdecydować jak z nim postąpić, czyli czy go zaakceptować, czy odrzucić.

Tradycyjny filtr pakietów można poprzez manipulowanie nagłówkami oszukać tak, aby akceptował pakiety, które powinien upuścić. Można tego dokonać poprzez manipulowanie flagami (zwłaszcza SYN) w nagłówkach pakietów TCP, tak by stworzyć złudzenie, że sfałszowany pakiet jest częścią już nawiązanego połączenia. Dzieje się tak ponieważ ten przestarzały sposób filtrowania nie daje możliwości śledzenia połączeń. Filtrowanie dynamiczne daje możliwość wykrycia takich pakietów i ich upuszczenia. Spowoduje ono również, że wszystkie próby "niewidocznych" skanowań (stealth scans) będą nieudane. Niewidoczne skanowanie to takie, które polega na wysłaniu przez skaner portów do skanowanego komputera serii pakietów z tak ustalonymi flagami, że szansa na ich zapisanie w logach jest mniejsza niż w przypadku zwykłych pakietów z flagą SYN.

Ponadto iptables posiada kilka innych bardzo użytecznych narzędzi, jak np. NAT (Network Adress Translation) czy limitowanie prędkości transmisji. To drugie jest szczególnie przydatne do zapobiegania atakom typu DoS (Denial of Service) typu SYN flood.

Połączenie TCP jest nawiązywane przez tak zwane potrójne uścisnięcie ręki. Podczas nawiązywania połączenia klient wysyła do serwera pakiet z ustawioną flagą SYN. Kiedy serwer otrzyma taki pakiet odpowiada na niego wysłaniem pakietu SYN+ACK. Kiedy ten pakiet dotrze do klienta odpowiada on trzecim pakietem, ACK, którym potwierdza nawiązanie połączenia.

Atak typu SYN flood polega na wysłaniu pakietu typu SYN i nie odpowiadaniu na pakiety SYN+ACK. Najczęściej wysyłane są pakiety ze sfałszowanym źródłowym adresem IP, ponieważ atakujący i tak nie oczekuje na nie odpowiedzi. Serwer po otrzymaniu każdego pakietu SYN dodaje odpowiedni wpis do kolejki pół-otwartych połączeń i rozpoczyna oczekiwanie na pakiet ACK. Ponieważ ilość miejsc w kolejce jest ograniczona możliwe jest jej przepelnienie i w konsekwencji uniemożliwienie nawiązywania

przez komputer zwykłych połączeń. Jeśli pakiet ACK nie przyjdzie po określonym okresie czasu wpis zostaje automatycznie usunięty z kolejki. Ten okres czasu zależy od atakowanego komputera, ale zwykle waha się pomiędzy 30 i 60 sekundami. Klient rozpoczyna atak przez wysłanie serii pakietów SYN z różnymi źródłowymi adresami IP i wysyła kolejne tak szybko i często jak to tylko możliwe, czym wypełnia kolejkę pół-otwartych połączeń i uniemożliwia innym klientom nawiązania połączenia z atakowanym komputerem.

I tu właśnie przydaje się limitowanie prędkości transmisji. Dzięki opcjom -m limit --limit 1/s można regulować ilość przyjmowanych pakietów SYN. Dzięki temu możliwe jest ograniczenie ilości pakietów SYN przyjmowanych w ciągu każdej sekundy co udaremnia wszystkie ataki typu SYN flood.

Uwaga: Inną metodą zapobiegania atakom typu SYN flood są SYN cookies, które pozwalają komputerowi na odpowiadanie na pakiety SYN bez zapelniania miejsca w kolejce połączeń. SYN cookies są już częścią jądra Linux i można z nich korzystać poprzez zaznaczenie w jego konfiguracji odpowiednich opcji, wciąż jest to jednak mocno eksperymentalny projekt.

A teraz pora na odrobinę praktyki.

Kiedy iptables zostaną załadowane do jądra powstanie 5 "zaczepów", na których można umieścić reguły. Ich nazwy to INPUT, OUTPUT, FORWARD, PREROUTING i POSTROUTING. Każdy z nich nazywany jest łańcuchem i składa się listy reguł. Każda z nich mówi o tym jak ma postępować firewall w przypadku otrzymania pakuetu z określonym nagłówkiem. Jeśli pakiet nie pasuje do żadnej z reguł jest traktowany zgodnie z polityką określona dla całego łańcucha, do którego trafił.

Własne reguły można umieszczać bezpośrednio w jednym z 5 głównych łańcuchów, bądź też dodać własne łańcuchy do któregoś z nich i dopiero w nich umieścić określone reguły.

Iptables posiadają następujące opcje:

Opcja:	Opis:
-A	Dodaj
-D	Usuń
-I	Umieść w określonym miejscu
-R	Zamień
-L	Pokaź listę
-F	Usuń wszystkie reguły z łańcucha
-Z	Wyzeruj liczniki łańcucha
-C	Sprawdź pakiet w łańcuchu
-N	Utwórz nowy, zdefiniowany przez użytkownika łańcuch
-X	Usuń zdefiniowany przez użytkownika łańcuch
-P	Zmień politykę łańcucha
-E	Zmień nazwę łańcucha
-p	Protokół
-s	Źródłowy adres/maska
-d	Docelowy adres/maska
-i	Interfejs, na którym pakiet został przyjęty
-o	Interfejs, przez który pakiet zostanie wysłany
-j	Cel
-m	Rozszerzone dopasowanie, przez podanie tekstu
-n	Numeryczne wynikowe adresy i porty
-t	Tabela do manipulowania
-v	Tryb szczegółowy
-x	Rozszerzanie numerów (podaje dokładne wartości)
-f	Dopasuj drugie i dalsze fragmenty
-V	Wersja pakietu
--line-numbers	Wyświetla numery linii na liście

Zaczniemy od prostego przykładu, upuszczania wszystkich pakietów ICMP przychodzących do naszego komputera.

Listing 1: Upuszczanie pakietów ICMP

```
# iptables -A INPUT -p icmp -j DROP
```

Na pierwszym miejscu podajemy łańcuch, do którego chcemy dopisać regułę, następnie protokół, do którego należą pakiety, a wreszcie cel. Celem może być nazwa zdefiniowanego przez użytkownika łańcucha, lub jedno ze specjalnych wyrażeń: ACCEPT, DROP, REJECT, LOG, QUEUE, lub MASQUERADE.

Uwaga: Cel LOG nie jest celem ostatecznym. Pakiet pasujący do reguły z takim celem zostanie również przetestowany pod kątem przynależności do innych reguł. Umożliwia to jednoczesne logowanie i przetwarzanie pakietów.

Sprawdzimy teraz skutki naszych działań. Wpisujemy polecenie ping localhost i czekamy na odpowiedź. Nie powinniśmy jej otrzymać ponieważ wszystkie pakiety ICMP przychodzące do naszego komputera są upuszczane. Uniemożliwiliśmy sobie również pingowanie innych komputerów, ponieważ także pakiety ICMP z odpowiedziami na nasze zapytania będą przez nasz firewall upuszczane. Wyczyśmy teraz wszystkie regułki i pozwólmy na swobodny przepływ pakietów ICMP.

Listing 2: Czyszczenie wszystkich reguł

```
# iptables -F
```

Przyjrzyjmy się dynamicznemu filtrowaniu pakietów w iptables. Jeśli chcemy włączyć dynamiczny nadzór nad pakiety przychodząymi na interfejs eth0 musimy ustalić następującą regułę:

Listing 3: Akceptowanie pakietów należących do już nawiązanego połączenia

```
# iptables -A INPUT -i eth0 -m state --state ESTABLISHED,RELATED -j ACCEPT
```

Wprowadzenie takiej reguły skutkuje akceptowaniem wszystkich przychodzących pakietów, które należą do już nawiązanych połączeń lub są z nimi powiązane. Jeśli przed dodaniem tej reguły wpiszemy dodatkowo polecenie iptables -A INPUT -i eth0 -m state --state INVALID -j DROP załadujemy rozszerzenie "state", dzięki któremu będziemy mieli możliwość upuszczania każdego pakietu, którego brakuje w tabeli stanu. Aby umożliwić innym łączению się z naszym komputerem możemy użyć flagi --state NEW. Iptables zawiera całą serię modułów o różnym przeznaczeniu, niektóre z nich to:

Rozszerzenie/Test	Opis	Rozszerzone opcje
mac	Test polegający na sprawdzaniu źródłowego adresu mac.	--mac-source
state	Test stanu	--state (stany to ESTABLISHED, RELATED, INVALID, NEW)
limit	Ograniczenie częstotliwości pasowania reguły	--limit, --limit-burst
owner	Próba sprawdzenia pewnych charakterystyk twórcy pakietu	--uid-owner userid --gid-owner groupid --pid-owner processid --sid-owner sessionid
unclean	Wykonuje różne losowe testy sprawdzające na pakiecie.	

Następne ćwiczenie będzie polegało na utworzeniu własnego łańcucha i przypisaniu go do jednego z już istniejących głównych łańcuchów.

Listing 4: Definiowanie własnego łańcucha

(Tworzymy nowy łańcuch z jedną regułą)

```
# iptables -X mychain
# iptables -N mychain
```

```
# iptables -A mychain -i eth0 -m state --state ESTABLISHED,RELATED -j ACCEPT
(Domyślna polityka polega na przepuszczaniu całego wychodzącego ruchu. Ruch przychodzący
będzie upuszczany.
# iptables -P OUTPUT ACCEPT
# iptables -P INPUT DROP
(Cały przychodzący ruch przekierowujemy do naszego łańcucha)
# iptables -A INPUT -j mychain
```

Dzięki dodaniu tych reguł zmienimy naszą politykę na następującą: wszystkie wychodzące pakiety będą przepuszczane, a pakiety przychodzące upuszczane.

Więcej informacji można znaleźć w [dokumentacji netfilter/iptables](#).

Pora na pełnowartościowy przykład. W tym przypadku zasady budowy naszego firewalla są następujące:

- Do firewalla można łączyć się tylko przez SSH (port 22).
- Lokalna sieć powinna mieć dostęp do HTTP, HTTPS i SSH (pozwalamy również na korzystanie z DNS).
- Część ruchu ICMP jest niepożądana, bo może posiadać niechcianą przez nas zawartość.
- Wszystkie skanowania portów powinny zostać wykryte i zalogowane.
- Należy uniemożliwić ataki typu SYN.
- Cała reszta niechcianych połączeń powinna być upuszczana i logowana.

Listing 5: /etc/init.d/firewall

```
#!/sbin/runscript
IPTABLES=/sbin/iptables
IPTABLESSAVE=/sbin/iptables-save
IPTABLESRESTORE=/sbin/iptables-restore
FIREWALL=/etc/firewall.rules
DNS1=212.242.40.3
DNS2=212.242.40.51
#inside
IIP=10.0.0.2
IINTERFACE=eth0
LOCAL_NETWORK=10.0.0.0/24
#outside
OIP=217.157.156.144
OINTERFACE=eth1

opts="${opts} showstatus panic save restore showoptions rules"

depend() {
    need net
}

rules() {
    stop
    ebegin "Setting internal rules"

    einfo "Setting default rule to drop"
    $IPTABLES -P FORWARD DROP
    $IPTABLES -P INPUT    DROP
    $IPTABLES -P OUTPUT   DROP

    #default rule
    einfo "Creating states chain"
    $IPTABLES -N allowed-connection
    $IPTABLES -F allowed-connection
    $IPTABLES -A allowed-connection -m state --state ESTABLISHED,RELATED -j ACCEPT
    $IPTABLES -A allowed-connection -i $IINTERFACE -m limit -j LOG --log-prefix \
        "Bad packet from ${IINTERFACE}:"
    $IPTABLES -A allowed-connection -j DROP

    #ICMP traffic
    einfo "Creating icmp chain"
```

```

$IPTABLES -N icmp_allowed
$IPTABLES -F icmp_allowed
$IPTABLES -A icmp_allowed -m state --state NEW -p icmp --icmp-type \
    time-exceeded -j ACCEPT
$IPTABLES -A icmp_allowed -m state --state NEW -p icmp --icmp-type \
    destination-unreachable -j ACCEPT
$IPTABLES -A icmp_allowed -p icmp -j LOG --log-prefix "Bad ICMP traffic:"
$IPTABLES -A icmp_allowed -p icmp -j DROP

#Incoming traffic
einfo "Creating incoming ssh traffic chain"
$IPTABLES -N allow-ssh-traffic-in
$IPTABLES -F allow-ssh-traffic-in
#Flood protection
$IPTABLES -A allow-ssh-traffic-in -m limit --limit 1/second -p tcp --tcp-flags \
    ALL RST --dport ssh -j ACCEPT
$IPTABLES -A allow-ssh-traffic-in -m limit --limit 1/second -p tcp --tcp-flags \
    ALL FIN --dport ssh -j ACCEPT
$IPTABLES -A allow-ssh-traffic-in -m limit --limit 1/second -p tcp --tcp-flags \
    ALL SYN --dport ssh -j ACCEPT
$IPTABLES -A allow-ssh-traffic-in -m state --state RELATED,ESTABLISHED -p tcp --dport
ssh -j ACCEPT

#outgoing traffic
einfo "Creating outgoing ssh traffic chain"
$IPTABLES -N allow-ssh-traffic-out
$IPTABLES -F allow-ssh-traffic-out
$IPTABLES -A allow-ssh-traffic-out -p tcp --dport ssh -j ACCEPT

einfo "Creating outgoing dns traffic chain"
$IPTABLES -N allow-dns-traffic-out
$IPTABLES -F allow-dns-traffic-out
$IPTABLES -A allow-dns-traffic-out -p udp -d $DNS1 --dport domain \
    -j ACCEPT
$IPTABLES -A allow-dns-traffic-out -p udp -d $DNS2 --dport domain \
    -j ACCEPT

einfo "Creating outgoing http/https traffic chain"
$IPTABLES -N allow-www-traffic-out
$IPTABLES -F allow-www-traffic-out
$IPTABLES -A allow-www-traffic-out -p tcp --dport www -j ACCEPT
$IPTABLES -A allow-www-traffic-out -p tcp --dport https -j ACCEPT

#Catch portscanners
einfo "Creating portscan detection chain"
$IPTABLES -N check-flags
$IPTABLES -F check-flags
$IPTABLES -A check-flags -p tcp --tcp-flags ALL FIN,URG,PSH -m limit \
    --limit 5/minute -j LOG --log-level alert --log-prefix "NMAP-XMAS:"
$IPTABLES -A check-flags -p tcp --tcp-flags ALL FIN,URG,PSH -j DROP
$IPTABLES -A check-flags -p tcp --tcp-flags ALL ALL -m limit --limit \
    5/minute -j LOG --log-level 1 --log-prefix "XMAS:"
$IPTABLES -A check-flags -p tcp --tcp-flags ALL ALL -j DROP
$IPTABLES -A check-flags -p tcp --tcp-flags ALL SYN,RST,ACK,FIN,URG \
    -m limit --limit 5/minute -j LOG --log-level 1 --log-prefix "XMAS-PSH:"
$IPTABLES -A check-flags -p tcp --tcp-flags ALL SYN,RST,ACK,FIN,URG -j DROP
$IPTABLES -A check-flags -p tcp --tcp-flags ALL NONE -m limit \
    --limit 5/minute -j LOG --log-level 1 --log-prefix "NULL_SCAN:"
$IPTABLES -A check-flags -p tcp --tcp-flags ALL NONE -j DROP
$IPTABLES -A check-flags -p tcp --tcp-flags SYN,RST SYN,RST -m limit \
    --limit 5/minute -j LOG --log-level 5 --log-prefix "SYN/RST:"
$IPTABLES -A check-flags -p tcp --tcp-flags SYN,RST SYN,RST -j DROP
$IPTABLES -A check-flags -p tcp --tcp-flags SYN,FIN SYN,FIN -m limit \
    --limit 5/minute -j LOG --log-level 5 --log-prefix "SYN/FIN:"
$IPTABLES -A check-flags -p tcp --tcp-flags SYN,FIN SYN,FIN -j DROP

```

```

# Apply and add invalid states to the chains
einfo "Applying chains to INPUT"
$IPTABLES -A INPUT -m state --state INVALID -j DROP
$IPTABLES -A INPUT -j icmp_allowed
$IPTABLES -A INPUT -j check-flags
$IPTABLES -A INPUT -i lo -j ACCEPT
$IPTABLES -A INPUT -j allow-ssh-traffic-in
$IPTABLES -A INPUT -j allowed-connection

einfo "Applying chains to FORWARD"
$IPTABLES -A FORWARD -m state --state INVALID -j DROP
$IPTABLES -A FORWARD -j icmp_allowed
$IPTABLES -A FORWARD -j check-flags
$IPTABLES -A FORWARD -o lo -j ACCEPT
$IPTABLES -A FORWARD -j allow-ssh-traffic-in
$IPTABLES -A FORWARD -j allow-www-traffic-out
$IPTABLES -A FORWARD -j allowed-connection

einfo "Applying chains to OUTPUT"
$IPTABLES -A OUTPUT -m state --state INVALID -j DROP
$IPTABLES -A OUTPUT -j icmp_allowed
$IPTABLES -A OUTPUT -j check-flags
$IPTABLES -A OUTPUT -o lo -j ACCEPT
$IPTABLES -A OUTPUT -j allow-ssh-traffic-out
$IPTABLES -A OUTPUT -j allow-dns-traffic-out
$IPTABLES -A OUTPUT -j allow-www-traffic-out
$IPTABLES -A OUTPUT -j allowed-connection

#Allow client to route through via NAT (Network Address Translation)
$IPTABLES -t nat -A POSTROUTING -o $IINTERFACE -j MASQUERADE
eend $?

}

start() {
ebegin "Starting firewall"
if [ -e "${FIREWALL}" ]; then
    restore
else
    einfo "${FIREWALL} does not exists. Using default rules."
    rules
fi
eend $?
}

stop() {
ebegin "Stopping firewall"
$IPTABLES -F
$IPTABLES -t nat -F
$IPTABLES -X
$IPTABLES -P FORWARD ACCEPT
$IPTABLES -P INPUT ACCEPT
$IPTABLES -P OUTPUT ACCEPT
eend $?
}

showstatus() {
ebegin "Status"
$IPTABLES -L -n -v --line-numbers
einfo "NAT status"
$IPTABLES -L -n -v --line-numbers -t nat
eend $?
}

panic() {
ebegin "Setting panic rules"
$IPTABLES -F
}

```

```

$IPTABLES -X
$IPTABLES -t nat -F
$IPTABLES -P FORWARD DROP
$IPTABLES -P INPUT DROP
$IPTABLES -P OUTPUT DROP
$IPTABLES -A INPUT -i lo -j ACCEPT
$IPTABLES -A OUTPUT -o lo -j ACCEPT
eend $?
}

save() {
ebegin "Saving Firewall rules"
$IPTABLESSAVE > ${FIREWALL}
eend $?
}

restore() {
ebegin "Restoring Firewall rules"
$IPTABLESRESTORE < ${FIREWALL}
eend $?
}

restart() {
svc_stop; svc_start
}

showoptions() {
echo "Usage: $0 {start|save|restore|panic|stop|restart|showstatus}"
echo "start)      will restore setting if exists else force rules"
echo "stop)       delete all rules and set all to accept"
echo "rules)      force settings of new rules"
echo "save)       will store settings in ${FIREWALL}"
echo "restore)    will restore settings from ${FIREWALL}"
echo "showstatus) Shows the status"
}

```

Kilka rad dotyczących tworzenia firewalli:

1. Warto spisać zasady firewalla zanim przystąpi się do ich implementowania
2. Nie warto nadmiernie komplikować zestawu reguł
3. Należy poznać podstawy pracy każdego z protokołów, np. poprzez przeczytanie odpowiednich [RFC](#)
4. Należy pamiętać, że firewall jest kolejnym programem uruchamianym z przywilejami roota
5. Warto przetestować firewall zanim na dobre zacznie się go używać

W przypadku, gdy iptables wydają się zbyt skomplikowane aby je zrozumieć, lub też ich skonfigurowanie zajmowałoby zbyt dużo czasu można skorzystać z programu Shorewall. Jest to prosty w konfiguracji firewall zabezpieczający komputer przy pomocy iptables.

12.f. Squid

Squid to doskonały serwer proxy, dzięki któremu możemy filtrować ruch sieciowy w oparciu o czas, wyrażenia regularne, źródłowy lub docelowy adres IP, domenę, przeglądarkę, uwierzytelniane konta użytkowników, typ MIME oraz numer portu (wraz z protokołem).

W poniższym przykładzie opisuję odfiltrowywanie bannerów zamiast filtrowania treści pornograficznych. Robię tak ponieważ chcę uniknąć sytuacji, że Gentoo.org zostanie zaindeksowane jako strona zawierająca treści porno, a poza tym nie mam czasu na szukanie dla was ciekawych stron.

Moja zasady wyglądają następująco:

- Na surfowanie w Internecie (HTTP/HTTPS) zezwalamy jedynie w godzinach pracy (poniedziałek-piątek 8-17, soboty 8-13). Jeśli pracownicy zostają po godzinach powinni zajmować się pracą, a nie chodzeniem po Internecie.
- Pobieranie plików nie jest dozwolone (żadnych .exe, .com, .arj, .zip, .ASF, .avi, .mpg, .mpeg, etc.)
- Nie lubimy bannerów, więc będą one odfiltrowywane i zastępowane przezroczystym gifem. (i to jest to miejsce, w którym trzeba uruchomić wyobraźnię...)
- Wszystkie inne połączenie do i z Internetu są zabronione.

Proces implementowania tych reguł składa się z 4 prostych kroków.

Listing 6: /etc/squid/squid.conf

```

# Bind to a ip and port
http_port 10.0.2.1:3128

# Standard configuration
hierarchy_stoplist cgi-bin ?
acl QUERY urlpath_regex cgi-bin \?
no_cache deny QUERY

# Add basic access control lists
acl all src 0.0.0.0/0.0.0.0
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255

# Add who can access this proxy server
acl localnet src 10.0.0.0/255.255.0.0

# And ports
acl SSL_ports port 443
acl Safe_ports port 80
acl Safe_ports port 443
acl purge method PURGE

# Add access control list based on regular
# expressions within urls
acl archives urlpath_regex "/etc/squid/files.acl"
acl url_ads url_regex "/etc/squid/banner-ads.acl"

# Add access control list based on time and day
acl restricted_weekdays time MTWTF 8:00-17:00
acl restricted_weekends time A 8:00-13:00

acl CONNECT method CONNECT

#allow manager access from localhost
http_access allow manager localhost
http_access deny manager

# Only allow purge requests from localhost
http_access allow purge localhost
http_access deny purge

# Deny requests to unknown ports
http_access deny !Safe_ports

# Deny CONNECT to other than SSL ports
http_access deny CONNECT !SSL_ports

# My own rules

# Add a page do be displayed when
# a banner is removed
deny_info NOTE_ADS_FILTERED url_ads

# Then deny them
http_access deny url_ads

# Deny all archives
http_access deny archives

# Restrict access to work hours
http_access allow localnet restricted_weekdays
http_access allow localnet restricted_weekends

```

```
# Deny the rest
http_access deny all
```

Następnie wypełniamy listę plików, na których pobieranie nie pozwalamy naszym użytkownikom. Na liście są pliki zip, viv, exe, mp3, rar, ace, avi, mov, mpg, mpeg, au, ra, arj, tar, gz i z.

Listing 7: /etc/squid/files.acl

```
\\. [Zz] [Ii] [pP] $
\\. [Vv] [Ii] [Vv] .*
\\. [Ee] [Xx] [Ee] $
\\. [Mm] [Pp] 3$
\\. [Rr] [Aa] [Rr] $
\\. [Aa] [Cc] [Ee] $
\\. [Aa] [Ss] [Ff] $
\\. [Aa] [Vv] [Ii] $
\\. [Mm] [Oo] [Vv] $
\\. [Mm] [Pp] [Gg] $
\\. [Mm] [Pp] [Ee] [Gg] $
\\. [Aa] [Uu] $
\\. [Rr] [Aa] $
\\. [Aa] [Rr] [Jj] $
\\. [Tt] [Aa] [Rr] $
\\. [Gg] [Zz] $
\\. [Zz] $
```

Uwaga: Warto zwrócić uwagę na nawiasy [] zawierające małą i dużą literę. Dzięki temu nikt nie oszuka naszego filtra poprzez wpisanie rozszerzenia pliku w wersji AvI zamiast avi.

Następnie dodamy wyrażenia regularne dzięki którym będziemy identyfikować bannery. Spodziewam się, że okażecie się tu znacznie bardziej kreatywni niż ja.

Listing 8: /etc/squid/banner-ads.acl

```
/adv/.*\.\.gif$
/[Aa]ds/.*\.\.gif$
/[Aa]d[Pp]ix/
/[Aa]d[Ss]erver
/[Aa] [Dd]/.*\.[GgJj] [IiPp] [FfGg] $
/[Bb]annerads/
/adbanner.*\.[GgJj] [IiPp] [FfGg] $
/images/ad/
/reklame/
/RealMedia/ads/.*
^http://www\.\.submit-it.*
^http://www\.\.eads.*
^http://ads\.
^http://ad\.
^http://ads02\.
^http://adaver.*\.
^http://adforce\.
adbot\.\.com
/ads/.*\.\.gif.*
ad\..\*cgi
/Banners/
/SmartBanner/
/Ads/Media/Images/
^http://static\.\.wired\.\.com/advertising/
^http://*\.\.dejanews\.\.com/ads/
^http://adfu\.\.blockstackers\.\.com/
^http://ads2\.\.zdnet\.\.com/adverts
^http://www2\.\.burstnet\.\.com/gifs/
```

```
^http://www.\.valueclick\.com/cgi-bin/cycle
^http://www.\.altavista\.com/av/gifs/ie_horiz\.gif
```

Ostatnią częścią całego procesu konfigurowania serwera proxy jest przygotowanie pliku, który będzie wyświetlany zamiast usuniętego bannera. Jest to prosty plik html zawierający plik gif o rozmiarach 4x4.

Listing 9: /etc/squid/errors/NOTE_ADS_FILTERED

```
<HTML>
<HEAD>
<META HTTP-EQUIV="REFRESH" CONTENT="0; URL=http://localhost/images/4x4.gif">
<TITLE>ERROR: The requested URL could not be retrieved</TITLE>
</HEAD>
<BODY>
<H1>Add filtered!</H1>
```

Uwaga: Nie należy zamykać tagów <HTML><BODY>. Squid zrobi to za nas.

Jak widać Squid posiada bardzo szerokie możliwości i jest bardzo efektywny zarówno jako filtr treści jak i serwer proxy. W dużych sieciach może z łatwością współpracować z innymi serwerami proxy. Moja konfiguracja najlepiej funkcjonuje w małych sieciach, składających się z góra 20 użytkowników.

Polecam używanie filtra pakietów (iptables) wraz z bramą aplikacyjną (Squid) nawet jeśli serwer proxy znajduje się w bezpiecznym miejscu i nikt z zewnątrz nie ma do niego dostępu. Zawsze istnieje możliwość ataku od wewnętrz.

Ostatnia kwestia to odpowiednie skonfigurowanie przeglądarek użytkowników, tak aby korzystały z naszego serwera proxy. Brama uniemożliwi użytkownikom jakkolwiek kontakt ze światem zewnętrznym sposobami innymi niż przez serwer proxy.

Uwaga: W Mozilli dokonuje się tego w menu Edit->Preferences->Advanced->Proxies.

Można to również robić w sposób niewidoczny dla użytkownika (transparentny) poprzez przekazanie przy pomocy iptables całego wychodzącego ruchu do serwera proxy. Można tego dokonać poprzez dodanie do łańcuchów forwarding/prerouting następującej reguły:

Listing 10: Konfigurowanie przekazywania portów do serwera proxy

```
# iptables -t nat -A PREROUTING -p tcp --dport 80 -j DNAT --to proxyhost:3128
# iptables -t nat -A PREROUTING -p tcp --dport 443 -j DNAT --to proxyhost:3128
```

Uwaga: Jeśli proxy działa na tym samym komputerze co filtr pakietów (nie polecam takiego rozwiązania, ale czasem z braku dodatkowych komputerów jest to konieczne) należy użyć celu REDIRECT zamiast DNAT (REDIRECT kieruje ruch na localhost).

12.g. Czego się nauczyliśmy?

Nauczyliśmy się, że:

1. Zimplementowanie firewalla może się wiązać z pewnym ryzykiem. Posiadanie źle skonfigurowanego firewalla jest gorsze niż nie posiadanie go w ogóle.
2. Nauczyliśmy się konfigurowania transparentnego proxy.
3. Dowiedzieliśmy się, że warunkiem budowy solidnego firewalla jest dobra znajomość protokołów sieciowych.
4. Wiemy, że ruch IP nie zawsze zawiera pożądane dane, np. pakiety ICMP mogą zawierać szkodliwe informacje.
5. Umiemy zapobiegać atakom typu SYN.
6. Wiemy jak filtrować ruch HTTP, tak aby uniemożliwić pobieranie wirusów i niestosownych treści.
7. Umiemy połączyć filtr pakietów z bramą aplikacyjną co zapewnia nam lepszą kontrolę nad naszą siecią.

Teraz, pod warunkiem, że jest tonaprawdę niezbędne, należy przystąpić do konfigurowania własnego, dostosowanego do indywidualnych potrzeb firewalla.

13. Wykrywanie włamań

13.a. AIDE (Advanced Intrusion Detection Environment)

AIDE jest systemem wykrywania włamań będącym darmową alternatywą dla Tripwire (osoby dobrze znające Tripwire nie powinny mieć żadnych problemów z konfiguracją AIDE). Systemów HID używa się do wykrywania zmian w ważnych systemowych plikach konfiguracyjnych i binarnych. Dokonuje się tego poprzez utworzenie i zachowanie unikalnych hashy każdego z tych plików. Co jakiś czas, zwykle codziennie, przechowywane w bezpiecznym miejscu hashe są porównywane z nowo utworzonymi hashami odpowiednich plików, co umożliwia wykrycie dokonanych w nich zmian. HIDS jest więc doskonałym sposobem wykrywania wszelkiej niepożądanej aktywności w naszym systemie, a jego jedyną wadą jest to, że odpowiednie zaimplementowanie takiego systemu wymaga trochę pracy.

Plik konfiguracyjny jest oparty na wyrażeniach regularnych, makrach i regułach dla plików oraz katalogów. Mamy do dyspozycji następujące makra:

Makro Opis

ifdef	"Jeśli zdefiniowano"
ifndef	"Jeśli nie zdefiniowano..."
define	Definiuje zmienną
undef	Usuwa zmienną
ifhost	Jeśli "nazwa hosta"...
ifnhost	Jeśli nazwa hosta różna od "nazwa hosta"...
endif	Endif używa się po każdym z powyższych makr, poza define i undef.

Składnia

@@ifdef "nazwa"
@@ifndef "nazwa"
@@define "zmienna" "wartość"
@@undef "nazwa"
@@ifhost "nazwa hosta"
@@ifnhost "nazwa hosta"
@@endif

Makra te są bardzo pomocne kiedy administruje się więcej niż jednym komputerem z Gentoo i chce się używać AIDE na każdym z nich. Należy jednak zwrócić uwagę, że nie wszystkie komputery posiadają te same usługi i identyczne konta użytkowników.

Kolejny zestaw wyrażeń to flagi dla plików i katalogów, czyli kombinacja praw dostępu, właściwości plików i hashy (sum kontrolnych).

Flaga Opis

p	prawa dostępu
i	inoda
n	liczba połączeń
u	użytkownik
g	grupa
s	rozmiar
b	ilość bloków
m	mtime
a	atime
c	ctime
S	sprawdza przyrost rozmiaru
md5	suma kontrolna md5
sha1	suma kontrolna sha1
rmd160	suma kontrolna rmd160
tiger	suma kontrolna tiger
R	$p+i+n+u+g+s+m+c+md5$
L	$p+i+n+u+g$
E	Pusta grupa
>	Rosnący plik z logami $p+u+g+i+n+S$

Ponadto jeśli AIDE zostanie skompilowane ze wsparciem dla mhash dostępne będą dodatkowo takie opcje:

Flaga Opis

haval	suma kontrolna haval
gost	suma kontrolna gost
crc32	suma kontrolna crc32

Przy użyciu powyższych flag można stworzyć własny zestaw reguł. Na przykład:

Listing 1: Zbór reguł dla AIDE

```
All=R+a+sha1+rmd160
Norm=s+n+b+md5+sha1+rmd160
```

Ostatnią rzeczą jaką trzeba dodać do pliku konfiguracyjnego są zasady obsługi plików i katalogów. Regułę dodajemy poprzez dopisanie nazwy pliku bądź katalogu wraz z odpowiednim zestawem flag. AIDE doda wszystkie możliwe pliki rekursively o ile nie nakażemy inaczej.

Flaga Opis

- !** Nie dodawać tego pliku bądź katalogu.
- =** Dodaj ten katalog, ale nie rekursively.

A oto przykładowy końcowy efekt:

Listing 2: /etc/aide/aide.conf

```
@@ifndef TOPDIR
@@define TOPDIR /
@@endif

@@ifndef AIDEDIR
@@define AIDEDIR /etc/aide
@@endif

@@ifhost smbserv
@@define smbactive
@@endif

# The location of the database to be read.
database=file:@@{AIDEDIR}/aide.db

# The location of the database to be written.
database_out=file:aide.db.new

verbose=20
report_url=stdout

# Rule definition
All=R+a+sha1+rmd160
Norm=s+n+b+md5+sha1+rmd160

@@{TOPDIR} Norm
!@@{TOPDIR}etc/aide
!@@{TOPDIR}dev
!@@{TOPDIR}proc
!@@{TOPDIR}root
!@@{TOPDIR}tmp
!@@{TOPDIR}var/log
!@@{TOPDIR}var/run
!@@{TOPDIR}usr/portage
@@ifdef smbactive
!@@{TOPDIR}etc/smb/private/secrets.tdb
@@endif
=@@{TOPDIR}home Norm
```

W powyższym przykładzie definiujemy poprzez makra katalog główny oraz miejsce gdzie zainstalowano AIDE. AIDE sprawdza plik /etc/aide/aide.db przy kontrolowaniu integralności plików, ale podczas aktualniania lub tworzenia nowej bazy wszystkie informacje zapisuje do pliku /etc/aide/aide.db.new. Robimy tak, aby zapobiec automatyczнемu nadpisaniu starego pliku z bazą danych.

Opcja report_URL nie została jeszcze zaimplementowana, autor zamierza dzięki niej umożliwić programowi wysyłanie e-maili oraz wykonywanie skryptów.

AIDE w Gentoo jest instalowane wraz z działającym w większości sytuacji plikiem konfiguracyjnym oraz dwoma skryptami, skryptem pomagającym w konfiguracji oraz skryptem do crontabu. Pierwszy skrypt udostępnia użytkownikowi interfejs, dzięki któremu cały proces konfiguracji staje się znacznie przyjemniejszy. Listę wszystkich jego parametrów można uzyskać za pomocą polecenia aideinit --help. Na początek powinno wystarczyć aideinit -i, po chwili skrypt crontab odkryje bazę danych i będzie wysyłał e-maile tak jak zostało to tam skonfigurowane. Radzimy zapoznać się z zawartością pliku /etc/aide/aide.conf i sprawdzić czy wszystkie ustawienia w nim zapisane są odpowiednie dla określonej sytuacji.

Uwaga: Zadanie to może zająć trochę czasu, w zależności od sprzętu i użytych flag.

Uwaga: Jeśli zapomni się o ustawieniu aliasu pocztowego, który będzie przekierowywał całą pocztę roota na prywatne konto administratora to nigdy nie ujrzy się żadnego z raportów AIDE.

Przechowywanie plików bazy danych na tym samym systemie, na którym dokonujemy testów wiąże się z pewnym ryzykiem. Jeśli włamywacz będzie świadom tego, że używamy AIDE zwykle spróbuje zmienić plik bazy danych, uaktualnić go po wprowadzeniu zmian lub nawet zmodyfikować plik /usr/bin/aide. W związku z tym warto rozważyć nagrywanie każdej kolejnej wersji pliku z bazą oraz plików binarnych AIDE na płytę CD.

Więcej informacji znajduje się na stronach projektu [AIDE](#).

13.b. Snort

Snort jest to sieciowy system wykrywania włamań (NIDS - Network Intrusion Detection System). Aby go zainstalować i skonfigurować należy wykonać następujące czynności:

Listing 3: /etc/conf.d/snort.conf

```
PIDFILE=/var/run/snort_eth0.pid
MODE="full"
NETWORK="10.0.0.0/24"
LOGDIR="/var/log/snort"
CONF=/etc/snort/snort.conf
SNORT_OPTS="-D -s -u snort -dev -l $LOGDIR -h $NETWORK -c $CONF"
```

Listing 4: /etc/snort/snort.conf

```
(Krok 1)
var HOME_NET 10.0.0.0/24
var EXTERNAL_NET any
var SMTP $HOME_NET
var HTTP_SERVERS $HOME_NET
var SQL_SERVERS $HOME_NET
var DNS_SERVERS [10.0.0.2/32,212.242.40.51/32]
var RULE_PATH ./

(Krok 2)
preprocessor frag2
preprocessor stream4: detect_scans detect_state_problems detect_scans disable_evasion_alerts
preprocessor stream4_reassemble: ports all
preprocessor http_decode: 80 8080 unicode iis_alt_unicode double_encode iis_flip_slash full_whois
preprocessor rpc_decode: 111 32771
preprocessor bo: -nobrute
preprocessor telnet_decode

(Krok 3)
include classification.config

(Krok 4)
include $RULE_PATH/bad-traffic.rules
include $RULE_PATH/exploit.rules
include $RULE_PATH/scan.rules
include $RULE_PATH/finger.rules
include $RULE_PATH/ftp.rules
```

```

include $RULE_PATH/telnet.rules
include $RULE_PATH/smtp.rules
include $RULE_PATH/rpc.rules
include $RULE_PATH/rservices.rules
include $RULE_PATH/dos.rules
include $RULE_PATH/ddos.rules
include $RULE_PATH/dns.rules
include $RULE_PATH/tftp.rules
include $RULE_PATH/web-cgi.rules
include $RULE_PATH/web-coldfusion.rules
include $RULE_PATH/web-iis.rules
include $RULE_PATH/web-frontpage.rules
include $RULE_PATH/web-misc.rules
include $RULE_PATH/web-attacks.rules
include $RULE_PATH/sql.rules
include $RULE_PATH/x11.rules
include $RULE_PATH/icmp.rules
include $RULE_PATH/netbios.rules
include $RULE_PATH/misc.rules
include $RULE_PATH/attack-responses.rules
include $RULE_PATH/backdoor.rules
include $RULE_PATH/shellcode.rules
include $RULE_PATH/policy.rules
include $RULE_PATH/porn.rules
include $RULE_PATH/info.rules
include $RULE_PATH/icmp-info.rules
include $RULE_PATH/virus.rules
# include $RULE_PATH/experimental.rules
include $RULE_PATH/local.rules

```

Listing 5: /etc/snort/classification.config

```

config classification: not-suspicious,Not Suspicious Traffic,3
config classification: unknown,Unknown Traffic,3
config classification: bad-unknown,Potentially Bad Traffic, 2
config classification: attempted-recon,Attempted Information Leak,2
config classification: successful-recon-limited,Information Leak,2
config classification: successful-recon-largescale,Large Scale Information Leak,2
config classification: attempted-dos,Attempted Denial of Service,2
config classification: successful-dos,Denial of Service,2
config classification: attempted-user,Attempted User Privilege Gain,1
config classification: unsuccessful-user,Unsuccessful User Privilege Gain,1
config classification: successful-user,Successful User Privilege Gain,1
config classification: attempted-admin,Attempted Administrator Privilege Gain,1
config classification: successful-admin,Successful Administrator Privilege Gain,1

# NEW CLASSIFICATIONS
config classification: rpc-portmap-decode,Decode of an RPC Query,2
config classification: shellcode-detect,Executable code was detected,1
config classification: string-detect,A suspicious string was detected,3
config classification: suspicious-filename-detect,A suspicious filename was detected,2
config classification: suspicious-login,An attempted login using a suspicious username was detected,2
config classification: system-call-detect,A system call was detected,2
config classification: tcp-connection,A TCP connection was detected,4
config classification: trojan-activity,A Network Trojan was detected, 1
config classification: unusual-client-port-connection,A client was using an unusual port,2
config classification: network-scan,Detection of a Network Scan,3
config classification: denial-of-service,Detection of a Denial of Service Attack,2
config classification: non-standard-protocol,Detection of a non-standard protocol or event,2
config classification: protocol-command-decode,Generic Protocol Command Decode,3
config classification: web-application-activity,access to a potentially vulnerable web application,3
config classification: web-application-attack,Web Application Attack,1
config classification: misc-activity,Misc activity,3
config classification: misc-attack,Misc Attack,2
config classification: icmp-event,Generic ICMP event,3

```

```
config classification: kickass-porn, SCORE! Get the lotion!,1
```

Więcej informacji znajduje się na stronie projektu [Snort](#).

13.c. Wykrywanie nieprawidłowości przy pomocy chkrootkit.

HIDS taki jak AIDE to świetny sposób wykrywania niepożądanych zmian w systemie. Warto jednak mieć też drugą linię obrony. chkrootkit to program skanujący najważniejszy pliki systemowe pod kątem obecności rootkitów, czyli programów pisanych, aby ukryć obecność włamywacza i pozwolić mu na zachowanie dostępu do komputera. Program skanuje również system w poszukiwaniu śladów loggerów klawiszy i innych niepożądanych programów. Jest to bardzo użyteczne narzędzie, posiadające zresztą kilka podobnie działających alternatyw (jak rkhunter). Mimo wszystko należy pamiętać, że żaden z nich nie gwarantuje bezpieczeństwa systemu.

Najwygodniejszym sposobem korzystania z chkrootkit jako metody wykrywania włamań jest uruchamianie go przy pomocy crona. Przede wszystkim trzeba go jednak zainstalować - poleceniem emerge app-admin/chkrootkit. chkrootkit może być uruchamiany z linii poleceń przy pomocy komendy o tej samej nazwie, lub poprzez taki wpis w cronie:

Listing 6: Planownie chkrootkit jako zadania dla crona

```
0 3 * * * /usr/sbin/chkrootkit
```

14. Aktualizacje

14.a. Aktualizowanie

Udana instalacja systemu i jego odpowiednie zabezpieczenie to nie wszystko. Większość włamań do systemów udaje się dzięki wykorzystaniu znanych błędów w starych wersjach programów. Częste aktualizowanie systemu jest niezbędne dla zachowania odpowiedniego poziomu bezpieczeństwa.

Przede wszystkim należy się upewnić, że jest zainstalowana najnowsza wersja Portage. Jeśli ją posiadamy wystarczy wydać polecenie emerge --sync i zsynchronizować drzewo Portage, a następnie poleciem glsa-check --list sprawdzić czy system jest aktualny pod względem bezpieczeństwa.

Listing 1: Przykładowy wynik glsa-check -l

```
# glsa-check -l
WARNING: This tool is completely new and not very tested, so it should not be
used on production systems. It's mainly a test tool for the new GLSA release
and distribution system, it's functionality will later be merged into emerge
and eqquery.
Please read http://www.gentoo.org/proj/en/portage/glsa-integration.xml
before using this tool AND before reporting a bug.

[A] means this GLSA was already applied,
[U] means the system is not affected and
[N] indicates that the system might be affected.

200406-03 [N] sitecopy: Multiple vulnerabilities in included libneon ( net-misc/sitecopy )
200406-04 [U] Mailman: Member password disclosure vulnerability ( net-mail/mailman )
....
```

Ostrzeżenie: Narzędzie glsa-check wciąż jest eksperymentalne warto więc sprawdzić również inne źródła informacji, zwłaszcza jeśli naszym głównym priorytetem jest maksymalne bezpieczeństwo.

Wszystkie linie zawierające [A] oraz [U] mogą zostać zignorowane ponieważ oznaczają, że system nie jest podatny na błąd opisany w określonym GLSA.

Ważne: Warto zauważyc, że używane zwykle polecenie emerge -vpuD world nie musi zaktualizować wszystkich potencjalnie niebezpiecznych pakietów. Należy zawsze korzystać z narzędzia glsa-check, które sprawdzi czy wszystkie błędy opisane w GLSA są naprawione.

Listing 2: Sprawdzanie zgodności z GLSA

```
(Sprawdzamy czy w systemie występują błędy opisane w GLSA)
# glsa-check -t all
WARNING: This tool is completely new and not very tested, so it should not be
used on production systems. It's mainly a test tool for the new GLSA release
and distribution system, it's functionality will later be merged into emerge
and eqquery.
Please read http://www.gentoo.org/proj/en/portage/glsa-integration.xml
before using this tool AND before reporting a bug.
```

This system is affected by the following GLSA:
200504-06
200510-08
200506-14
200501-35
200508-12
200507-16

```
(Sprawdzamy, które pakiety zostaną przebudowane)
# glsa-check -p $(glsa-check -t all)
(Częściowy wynik)
Checking GLSA 200504-06
The following updates will be performed for this GLSA:
```

```
app-arch/sharutils-4.2.1-r11 (4.2.1-r10)
*****
Checking Glsa 200510-08
The following updates will be performed for this Glsa:
media-libs/xine-lib-1.1.0-r5 (1.1.0-r4)

(Wprowadzanie rekommendowanych zmian)
# glsa-check -f $(glsa-check -t all)
```

Jeśli zaktualizowana została jedna z usług działających na serwerze, należy pamiętać o jej ponownym uruchomieniu.

Zalecamy również dbanie o aktualność jądra systemu.

Aby otrzymywać mail za każdym razem gdy wypuszczane jest Glsa należy najpierw zapisać się do listy mailingowej gentoo-announce. Instrukcje na temat dołączania do tej i innych list mailingowych można znaleźć na stronie [Gentoo Linux Mailing List Overview](#).

Świetnym źródłem informacji dotyczących bezpieczeństwa jest [lista mailingowa bugtraq](#).

15. Prelink w Gentoo

1. Wprowadzenie

Co to jest Prelink i do czego służy?

Większość aplikacji korzysta ze współdzielonych bibliotek, które muszą zostać załadowane do pamięci w czasie pracy programu, co za każdym razem wiąże się z koniecznością dynamicznego połączenia programu z potrzebnymi mu bibliotekami. W przypadku większości małych programów ten proces trwa bardzo krótko. Jednak w przypadku programów napisanych w języku C++ lub posiadających wiele zależności symbolicznych, proces dynamicznego łączenia może zająć sporo czasu.

W przeciętnym systemie biblioteki rzadko ulegają modyfikacjom, stąd kiedy program jest uruchamiany, czynności wykonywane podczas dynamicznego łączenia są zawsze jednakowe. Prelink wykorzystuje ten fakt i dokonuje dynamicznego łączenia tylko raz, po czym zapisuje wyniki na stałe w pliku wykonywalnym programu.

Prelinkowanie może skrócić czas uruchamiania programów. Przykładowo czas potrzebny do uruchomienia typowego programu dla środowiska KDE może zostać skrócony nawet o 50%. Jedynym obowiązkiem użytkownika jest ponowne uruchomienie polecenia prelink po każdej aktualizacji biblioteki, używanej przez prelinkowany program.

Ostrzeżenie: **Prelinkowanie nie działa na Hardened Gentoo. Dzieje się tak dlatego, że oba projekty próbują zmienić adresowanie bibliotek współdzielonych. Jednak prelinkowanie z opcją -R przydziela przypadkowe adresy współdzielonym bibliotekom, co w pewnym stopniu zapewnia odpowiedni poziom bezpieczeństwa**

Podsumowanie

- Prelinkowania dokonuje się przy użyciu programu o nazwie... prelink. Program ten modyfikuje plik wykonywalny tak, aby uruchamiał się szybciej.
- Jeżeli jedna z bibliotek wykorzystywanych przez prelinkowany program ulegnie modyfikacji, konieczne jest ponowne prelinkowanie programu. W przeciwnym wypadku utracony zostanie efekt przyspieszonego uruchamiania. W skrócie: za każdym razem, kiedy poprzez portage aktualizowany jest pakiet, który aktualizuje jakieś biblioteki, wszystkie programy ich używające muszą zostać ponownie prelinkowane.
- Modyfikacje dokonane przez polecenie prelink są w pełni odwracalne. Polecenie posiada funkcję undo.
- Aktualne wersje Portage potrafią rozpoznać zmiany w sumach MD5 i znacznikach czasowych plików wykonywalnych, dokonane w wyniku użycia prelink.
- Nie ma potrzeby dopisywania FEATURES="prelink" do pliku make.conf. Portage automatycznie używa prelinkowania, jeżeli znajdzie w systemie zainstalowane polecenie prelink.

2. Instalacja polecenia Prelink

Instalacja programu

Aby rozpocząć, należy zainstalować program prelink. Podczas jego instalacji wykonywana jest seria testów, które mają na celu określenie, czy korzystanie z prelinkowania na danym systemie będzie w pełni bezpieczne.

Listing 2.1: Instalacja polecenia Prelink

```
# emerge prelink
```

U części użytkowników emergowanie programu prelink kończy się błędem z powodu niepowodzenia podczas testowania systemu. Owe testy zostały wprowadzone ze względów bezpieczeństwa. Można je pominąć, jednak w takim przypadku zachowanie polecenia prelink może być nieobliczalne. Przyczyną niepowodzenia testów jest zwykle nieprawidłowa instalacja podstawowych pakietów systemowych, takich jak binutils, gcc i glibc. Instalacja tych pakietów w podanej kolejności może wpływać na poprawę sytuacji.

Uwaga: Wskazówka: W przypadku pojawiienia się błędu można ręcznie zainstalować polecenie prelink (./configure ; make ; make check). Jeżeli wystąpił błąd, zawsze można przejrzeć pliki *.log w katalogu zawierającym programy testujące. Z zawartości tych plików można wyłowić kilka przydatnych wskazówek, pomocnych przy usuwaniu błędu.

Jeżeli istnieje określona sekwencja kroków, w wyniku której na innym systemie również pojawia się identyczny błąd, należy wysłać ich opis na [Bugzilla Gentoo](#).

Przygotowanie systemu

Należy również zwrócić uwagę, żeby wśród flag w zmiennych CFLAGS/CXXFLAGS nie znajdowała się opcja -fPIC. W przeciwnym

wypadku konieczne będzie jej usunięcie i ponowne przebudowanie całego systemu.

Konfiguracja

Wywołanie env-update spowoduje wygenerowanie pliku /etc/prelink.conf, w którym zawarta jest lista plików przeznaczonych do prelinkowania.

Listing 2.2: Wywoływanie env-update

```
# env-update
```

Niestety nie jest możliwe prelinkowanie programów skompilowanych przy użyciu starych wersji binutils. Większość z tych programów pochodzi z pakietów aplikacji rozpuszczalnych wyłącznie w postaci binarnej, które instalowane są zwykle w katalogu /opt. Utworzenie następującego pliku spowoduje, że ww. programy zostaną pominięte podczas prelinkowania.

Listing 2.3: /etc/env.d/60prelink

```
PRELINK_PATH_MASK="/opt"
```

Uwaga: Można określić więcej katalogów do pominięcia poprzez oddzielenie ich nazw dwukropkami.

3. Prelinkowanie

Użycie polecenia Prelink

Do prelinkowania wszystkich programów w katalogach wymienionych w pliku /etc/prelink.conf można użyć polecenia:

Listing 3.1: Prelinkowanie wyszczególnionych programów

```
# prelink -amR
```

Ostrzeżenie: Jeżeli na Twoim systemie jest mało miejsca na dysku, prelinkowanie całego systemu może się skończyć obcięciem i tym samym zniszczeniem niektórych plików wykonywalnych, czego efektem będzie pad systemu. Należy wtedy użyć polecenia file albo readelf, aby szybko sprawdzić stan podejrzanej pliku. Najlepiej jednak przed prelinkowaniem sprawdzić ilość wolnego miejsca na dyskach, wywołując df -h.

Opis użytych parametrów:

- a "All": prelinkuje wszystkie programy.
- m Oszczędzaj pamięć wirtualną. Przydatne w przypadku prelinkowania większej ilości bibliotek.
- R Tryb losowy -- przyporządkowuje przypadkowe adresy co skutkuje wzrostem poziomu bezpieczeństwa poprzez zwiększenie odporności na ataki z wykorzystaniem przepelnienia bufora (buffer overflow).

Uwaga: Więcej parametrów i szczegółów dotyczących ich użycia można odnaleźć na stronie manuala: man prelink.

Prelinkowanie w cronie

Pakiet sys-devel/prelink-20060213 i późniejsze instalują zadania do wykonania przez cron w pliku /etc/cron.daily/prelink. Aby je aktywnić należy edytować plik /etc/conf.d/prelink. Dzięki temu prelinkowanie będzie uruchamiane codziennie, zaoszczędzając nam ręcznego wpisywania komend

Przyśpieszanie uruchamiania KDE po prelinkowaniu

Czas uruchamiania KDE może zostać znacznie zredukowany w wyniku prelinkowania. Aby dodatkowo skrócić czas uruchamiania KDE, można je skonfigurować tak, aby przestało korzystać z programu kdeinit, który po prelinkowaniu nie jest już potrzebny.

Aby tego dokonać, wystarczy umieścić wiersz KDE_IS_PRELINKED="true" w pliku /etc/env.d/*kdepaths*.

4. Znane problemy i ich rozwiązania

"Cannot prelink against non-PIC shared library" (Nie można prelinkować z biblioteką skompilowaną bez opcji PIC)

Przyczyną tego problemu jest niewłaściwie skompilowana biblioteka, podczas komplikacji której nie podano parametru `-fPIC`.

Oto lista bibliotek, które nie zostały poprawione, lub których poprawienie nie jest możliwe:

- Biblioteki w pakiecie wine i pochodnych (w tym winex). Prelinkowanie w tym wypadku i tak nie spowodowałoby przyspieszenia programów systemu MS Windows.
- Biblioteka `/usr/lib/liblavfile-1.6.so.0` z pakietu media-video/mjpegtools.
- Biblioteki OpenGL Nvidia, `/usr/lib/opengl/nvidia/lib/libGL.so.*`. Z powodu problemów z wydajnością zostały one skompilowane bez wsparcia dla PIC.

Jeżeli sprawcą problemu jest biblioteka nie ujęta na tej liście, prosimy zgłosić ten fakt, najlepiej wraz z łatą, dodającą flagę `-fPIC` do `CFLAGS` we właściwych miejscach.

Po prelinkowaniu systemu niektóre statyczne programy przestają działać.

W systemie opartym o glibc tak naprawdę nie istnieje coś takiego, jak w 100% statyczny program. Skompilowanie programu statycznego w stosunku do glibc nie oznacza, że program nie będzie posiadał innych zależności w stosunku do innych plików systemowych. Oto, jak problem wyjaśnił Dick Howell:

"Pomysł ze statycznie skompilowanymi programami ma na celu uniezależnienie ich od zainstalowanych w systemie bibliotek. Niestety w Linuksie oraz, jak sądzę, w innych systemach opartych o GLIBC, nie jest to do końca możliwe. Jest np. sprawa biblioteki "libnss" (Name Service Switch, przez niektórych nie wiedzieć czemu, nazywana Network Security System), która udostępnia interfejs dostępu do różnych baz danych uwierzytelniania, informacji sieciowych i wielu innych rzeczy. Ma to na celu uniezależnienie aplikacji od konfiguracji sieciowej konkretnego systemu. Pomysł jest niezły, jednak zmiany w GLIBC mogą powodować problemy z jej załadunkiem. Nie można statyczniełączyć biblioteki "libnss", ponieważ dla każdego systemu jest ona inaczej skonfigurowana. Problem wydaje się leżeć w statycznym linkowaniu innych bibliotek składowych GLIBC, np. "libpthread", "libm" oraz "libc", w wyniku czego powstają niekompatybilne odwołania do funkcji biblioteki "libnss"."

Prelink kończy z błędem "prelink: dso.c:306: fdopen_dso: Assertion `j == k' failed."

Jest to znany problem, przystępnie opisany [tutaj](#). Prelink nie radzi sobie z plikami wykonywalnymi, skompresowanymi programem UPX. Począwszy od prelink w wersji 20021213 jedynym sposobem ominięcia problemu jest nakazanie poleceniu prelink pominiecie takich programów. W rozdziale [Konfiguracja](#) opisano prosty sposób, w jaki można tego dokonać.

Używam grsecurity i mam wrażenie, że prelinkowanie nie działa.

Aby prelinkować programy w systemie używającym grsecurity, z opcją losowego generowania adresów bazowych w funkcji `mmap()`, konieczne jest WYŁĄCZENIE tej funkcji dla pliku `/lib/ld-2.3.*.so`. Można tego dokonać przy użyciu polecenia `chpax`, jednak plik nie może być danym momencie używany przez żaden program (najlepiej zrobić to uruchamiając system z płyty CD).

Prelink kończy pracę z błędem "prelink: Can't walk directory tree XXXX: Too many levels of symbolic links".

Błąd ten może oznaczać nadmierny poziom skomplikowania dowiązań symbolicznych. Występuje na przykład, gdy dwa dowiązania wskazują wzajemnie na siebie. Częstym powodem takiego problemu jest na przykład `/usr/lib/lib -> lib`. Aby naprawić dowiązania, można je zmienić ręcznie lub posłużyć się programem dostarczonym razem z pakietem `symlinks`:

Listing 4.1: Naprawianie dowiązań

```
# emerge symlinks
# symlinks -drv /
```

Więcej informacji na temat tego błędu można znaleźć w [Bugzilli](#) oraz na [forum Gentoo](#).

5. Podsumowanie

Prelinkowanie może drastycznie przyspieszyć uruchamianie wielu dużych aplikacji. W dodatku jest ono wspierane przez Portage. Prelinkowanie jest również bezpieczne, ponieważ zawsze można odwrócić ten proces dla dowolnego pliku wykonywalnego, jeżeli sprawia on kłopoty. Wystarczy pamiętać, aby po aktualizacji glibc lub innych bibliotek zawsze ponownie uruchomić prelink w celu uaktualnienia wszystkich programów. Powodzenia!

16. Instalacja i konfiguracja demona CRON

1. Cron - podstawy

Co robi cron?

Cron jest demonem, który uruchamia zaplanowane zadania w oparciu o dane przekazane przez polecenie crontab. Program wykonuje swoją pracę sprawdzając co minutę czy w plikach crontab użytkowników znajdują się jakieś zadania do zrealizowania.

Uwaga: Określenia crontab używamy zarówno jako nazwy listy zadań do wykonania, jak i nazwy polecenia służącego do edycji tej listy.

Cron w praktyce

W Portage znajdują się co najmniej trzy implementacje crona, spośród których można wybierać. Wszystkie posiadają podobny interfejs. Mianowicie, używają komendy crontab lub podobnej. Istnieje także pokrewne narzędzie zwane Anacron, którego przeznaczeniem jest współpraca z cronem na systemach, które nie działają w sposób ciągły.

Zależnością każdego z tych trzech cronów jest pakiet sys-process/cronbase. Może on wystarczyć do podstawowych zastosowań crona.

Zanim zaczniemy pracę z cronem, musimy wybrać jedną z wersji tego programu. Dla wygody przedstawiamy poniżej informacje o dostępnych implementacjach.

2. Który cron spełni nasze potrzeby?

Vixie cron

Vixie cron jest wersją posiadającą wiele możliwości i opartą na cronie z SysV. Każdy użytkownik posiada własny plik crontab oraz ma możliwość używania zmiennych środowiskowych w obrębie tego pliku. W przeciwieństwie do innych odmian crona, oferuje wsparcie dla SELinux i PAM. Wspiera mniej architektur niż DCron, ale więcej niż Fcron.

Cechy programu sys-process/vixie-cron:

- Wsparcie dla SELinux
- Wsparcie dla PAM /etc/security/limits.conf
- Ustawianie zmiennych środowiskowych w plikach crontab (PATH, SHELL, HOME, itp.)
- Każdy użytkownik może posiadać własny crontab, do którego dostęp kontrolowany jest za pomocą cron.allow i cron.deny

Cron Dillon'a

Dcron w zamierzeniach ma być prostą, elegancką i bezpieczną implementacją crona. Nie pozwala na wykorzystanie zmiennych środowiskowych w plikach crontab, zaś wszystkie zadania są uruchamiane z /bin/sh. Podobnie jak w Vixie cron, każdy użytkownik posiada własny plik crontab.

Cechy programu sys-process/dcron:

- Szybki, prosty i wolny od zbędnych dodatków
- Dostęp do crontab jest ograniczony do grupy cron, tzn. nie polega na zewnętrznych zasobach

Fcron

Fcron w zamierzeniach ma zastąpić programy Vixie cron i Anacron. Został zaprojektowany do pracy na systemach, które nie działają w sposób ciągły. Ponadto zawiera dodatkowe możliwości takie jak: uruchamianie zadań w zależności od tego czy spełnione są określone warunki, zdolność przypisywania priorytetów zadaniom, możliwość przypisania zadania do uruchomienia przy starcie systemu. Więcej informacji można znaleźć na [stronie domowej Fcrona](#).

Cechy programu sys-process/fcron:

- Zaprojektowany do pracy na systemach, które nie działają w sposób ciągły, np. może rozpoczęć zadanie po ponownym uruchomieniu systemu
- Ustawianie zmiennych środowiskowych i wielu innych opcji dla plików crontab
- Każdy użytkownik może posiadać własny crontab, do którego dostęp kontrolowany jest za pomocą cron.allow i cron.deny
- Rozszerzona składnia polecenia crontab, zawierająca obsługę wielu nowych możliwości

Anacron

Anacron nie jest demonem crona. Zazwyczaj pracuje w połączeniu z cronem. Wykonuje polecenia w odstępach czasu podawanych w dniach oraz nie czyni założeń o ciągłym działaniu systemu. Uruchamia zadania, które nie zostały wywołane podczas wyłączeń systemu. Anacron przeważnie jest uruchamiany przez demona cron raz dziennie.

3. Korzystanie z crona

Instalacja

Wybieramy tę wersję crona, która nam najbardziej odpowiada i emergujemy ją.

Listing 3.1: Instalacja crona

```
# emerge dcron
# /etc/init.d/dcron start
# rc-update add dcron default
```

Jeśli nie zainstalowaliśmy Fcrona, to opcjonalnie możemy zainstalować Anacrona.

Listing 3.2: Instalacja anacrona

```
# emerge anacron
# /etc/init.d/anacron start
# rc-update add anacron default
```

Plik crontab dla systemu

Poinstalacyjne komunikaty z niektórych wersji crona informują o konieczności uruchomienia polecenia crontab /etc/crontab. Plik /etc/crontab jest plikiem crontab dla systemu. Demon cron dzięki sys-process/cronbase może uruchamiać skrypty z plików /etc/cron.{daily, hourly, weekly, monthly}. Należy wspomnieć, że tylko vixie-cron automatycznie planuje zadania w /etc/crontab. Użytkownicy programów Dcron i Fcron muszą uruchomić crontab /etc/crontab, każdorazowo po wprowadzeniu zmian w /etc/crontab.

Warto zauważyć, że zadania zaplanowane w systemowym pliku crontab, nie zostaną wyświetcone na liście zaplanowanych zadań, tj. po wydaniu komendy crontab -l.

Oczywiście nie musimy używać żadnego z programów cron. Jeśli zdecydujemy się na Dcrona lub Fcrona, to NIE uruchamiamy crontab /etc/crontab. Jeśli wybierzemy vixie-crona, to powinniśmy zakomentować wszystkie linie w pliku /etc/crontab.

Listing 3.3: Wykomentowanie wszystkich linii w pliku /etc/crontab

```
# sed -i -e "s/^/#/" /etc/crontab
```

Nadawanie zaufanym użytkownikom dostępu do crona

Jeśli chcemy, by użytkownicy inni niż root mieli dostęp do demona cron, to sugerujemy przeczytanie tego akapitu. W przeciwnym przypadku można przejść do następnej sekcji, tj. planowania zadań.

Uwaga: Nadanie innemu użytkownikowi dostępu do crona nie powoduje, że będzie on mógł uruchamiać zaplanowane zadania jako root. Jeśli chcemy, by użytkownik miał możliwość edycji pliku crontab roota, to należy zapoznać się sudo. Więcej informacji na temat tego narzędzia znajdziemy w dokumencie Sudo i sudoers w Gentoo.

Jeśli chcemy nadać użytkownikowi prawo do korzystania z plików crontab, to niezależnie, który pakiet wybierzemy, musimy go najpierw dodać do grupy cron. Dla przykładu, jeśli chcemy dodać użytkownika yarel do grupy cron, to wpisujemy:

Listing 3.4: Dodawanie użytkownika do grupy cron

```
# gpasswd -a yarel cron
```

Uwaga: Efekt dodania użytkownika do grupy będzie widoczny dopiero po jego przelogowaniu.

Jeśli używamy **Dcrona**, to jest to wszystko co musimy zrobić, by nadać użytkownikowi prawo do korzystania z crontab. Użytkownicy Dcrona mogą przejść do następnej sekcji, tj. [planowanie zadań](#). Pozostali czytają dalej.

Jeśli korzystamy z **Fcrona**, to musimy zmienić pliki /etc/fcron/fcron.deny oraz /etc/fcron/fcron.allow. Najbezpieczniejszy sposób to odmówienie dostępu wszystkimi (all), w pliku /etc/fcron/fcron.deny, a następnie wymienienie w pliku /etc/fcron/fcron.allow użytkowników, którzy mają mieć dostęp.

Ważne: Jeśli nie istnieje żaden z plików: /etc/feron/fcron.allow i /etc/feron/fcron.deny, to wszyscy użytkownicy z grupy cron będą mieli możliwość użycia polecenia crontab. fcron instalowany jest ze standardowym plikiem fcron.allow, który pozwala wszystkim użytkownikom z grupy cron, na korzystanie z fcrontab.

Listing 3.5: Uprawnienia w pliku fcron.deny

```
all
```

Przypuśćmy, że mamy użytkownika yarel, który powinien mieć możliwość planowania własnych zadań. Dodajemy go do pliku /etc/feron/fcron.allow w następujący sposób:

Listing 3.6: Uprawnienia w pliku fcron.allow

```
yarel
```

Wybierając Vixie crona, będziemy chcieli wyedytować plik /etc/cron.allow.

Ważne: Warto wspomnieć, że jeśli istnieje plik /etc/cron.allow, to tylko wymienieni w nim użytkownicy z grupy cron, będą mieli dostęp do crona. Ale jeśli istnieje pusty plik /etc/cron.deny, to wszyscy użytkownicy z grupy cron będą posiadali prawo dostępu! Nie zostawiamy pustego pliku /etc/cron.deny, jeśli nie posiadamy pliku /etc/cron.allow.

Przykładowo, jeśli chcemy nadać użytkownikowi yarel prawo dostępu do crona, to umieszczamy go w pliku /etc/cron.allow następująco:

Listing 3.7: Uprawnienia w pliku /etc/cron.allow

```
yarel
```

Planowanie zadań

W każdym pakiecie proces edycji plików crontab wygląda inaczej, ale wszystkie wersje obsługują podstawowy zestaw poleceń: dodawanie i zastępowanie pliku crontab, edycja i usuwanie oraz wyświetlanie listy zadań z plików crontab. Następujące zestawienie pokazuje jak używać komend w poszczególnych pakietach.

Wersja	Edycja crontab	Usuwanie crontab	Nowy crontab	Lista prac crona
crontab	crontab -e	crontab -d [użytkownik]	crontab plik	crontab -l
fcron	fcrontab -e	fcrontab -r [użytkownik]	fcrontab plik	fcrontab -l
vixie-cron	crontab -e	crontab -r -u [użytkownik]	crontab plik	crontab -l

Uwaga: Jeśli nie podamy parametru polecenia "usuń", zostanie usunięty aktualny crontab danego użytkownika

Uwaga: Fcron tworzy także dowiązanie symboliczne do crontaba.

Zanim zaczniemy używać tych komend, musimy zrozumieć strukturę pliku crontab. Każda linia takiego pliku wymaga podania pięciu pól określających czas. W kolejności są to: minuty (0-59), godziny (0-23), dni miesiąca (1-31), miesiące (1-12) oraz dni tygodnia (0-7, 1 to poniedziałek, 0 i 7 to niedziela). Dni tygodni i miesiące mogą być określane za pomocą trzyliterowych angielskich skrótów, takich jak: mon, tue, jan, feb, itp. Każde pole może także zawierać zakres wartości (np. 1-5 dla poniedziałek-piątek), listę wartości rozdzieloną przecinkiem (np. 1,2,3 albo mon,tue,wed) lub zakres wartości z określonym krokiem (np. 1-6/2 dla 1,3,5).

Może to wyglądać na zagmatwane, ale kilka przykładów powinno rozwiać wątpliwości.

Listing 3.8: Przykłady

```
# Uruchamia /bin/false co minutę przez cały rok
* * * * * /bin/false

# Uruchamia /bin/false o 1:35 w pon, wt, sr i 4-tego każdego miesiąca
35 1 4 * mon-wed /bin/false

# Uruchamia /bin/true o 22:25 2. marca
25 22 2 3 * /bin/true

# Uruchamia /bin/false o 2:00 w każdy poniedziałek, środę i piątek
0 2 * * 1-5/2 /bin/false
```

Uwaga: Zauważmy jak należy określać pola dotyczące dni tygodnia i dni miesiąca, zanim zaczną odnosić się do tego samego dnia. Jeśli wprowadzimy * dla jednego z nich, drugie będzie miało wyższy priorytet. Wpisanie * dla obydwu pól oznacza po prostu każdy dzień.

Przećwiczmy to, czego się właśnie nauczyliśmy. Wykonamy kolejne kroki tworzenia nowych zaplanowanych zadań. Na początku tworzymy plik o nazwie crons.cron i wypełniamy go tak, by wyglądał następująco:

Listing 3.9: Edycja pliku crons.cron

```
$ nano crons.cron
# Minuty Godziny Dni Miesiące Dni tygodnia
10 3 1 1 * /bin/echo "Nie lubię crona."
30 16 * 1,2 * /bin/echo "Trocę lubię crona."
* * * 1-12/2 * /bin/echo "Naprawdę lubię crona."
```

Teraz możemy dodać plik crontab do systemu używając komendy "nowy wpis" z tabeli powyżej.

Listing 3.10: Nowy plik crontab

```
# crontab crons.cron
```

Uwaga: Tak naprawdę to nie zobaczymy wyniku działania powyższych komend echo, chyba że użyjemy przekierowania.

Weryfikację zaplanowanych zadań możemy przeprowadzić z użyciem odpowiedniej komendy listowania z tabeli powyżej.

Listing 3.11: Lista zaplanowanych zadań

```
# crontab -l
```

Powinna pojawić się lista przypominająca zawartość pliku crons.cron. Jeśli tak nie jest, to możliwe że zostało użyte niewłaściwe polecenie, by dodać nowy plik crontab.

Zgodnie z powyższym plikiem crontab, co minutę każdej godziny, każdego dnia w co drugim miesiącu, wykonane będzie polecenie echo "Naprawdę lubię crona.". Oczywiście robimy to, jeśli naprawdę lubimy crona. Każdego dnia stycznia oraz lutego o godzinie 16:30 zostanie wykonana komenda echo "Trocę lubię crona.". Pierwszego stycznia o 3:10 zostanie wykonana komenda echo "Nie lubię crona.".

Użytkownicy Anacrona powinni czytać dalej, natomiast pozostali mogą przejść do następnego akapitu, tj. [edykcji plików crontab](#).

Użytkownicy Anacrona będą modyfikować zawartość pliku /etc/anacrontab. W pliku tym znajdują się cztery pola: liczba dni między kolejnymi uruchomieniami, opóźnienie w minutach, po którym zadanie jest uruchamiane, nazwa zadania oraz komenda, która będzie uruchamiana.

Przykładowo, by wywoływać co 5 dni, 10 minut po uruchomieniu Anacrona polecenie echo "Lubię anacrona." wpisujemy:

Listing 3.12: /etc/anacrontab

```
5 10 wasting-time /bin/echo "Lubię anacrona."
```

Program Anacron kończy działanie po tym, gdy wszystkie zadania wymienione w pliku anacrontab, zostaną wykonane. Jeśli chcemy sprawdzać te zadania codziennie, to musimy użyć crona. Instrukcja pod koniec następnego akapitu, wyjaśnia jak to osiągnąć.

Edycja plików crontab

Będźmy poważni, nie chcemy, by system co minutę mówił nam jak bardzo lubimy crona. Kolejny krok w poznawaniu tego programu, to usunięcie pliku crontab przy użyciu odpowiedniej komendy do usuwania z wymienionej powyżej tabeli. Po usunięciu pliku wylistujemy zaplanowane zadania, by upewnić się, że komenda do usuwania zadziałała.

Listing 3.13: Usuwanie pliku crontab

```
# crontab -d  
# crontab -l
```

Po wydaniu polecenia crontab -l nie powinna wyświetlać się lista zaplanowanych zadań. Jeśli pojawiają się wpisy, oznacza to, że polecenie do usuwania zadań nie zadziałało. Należy upewnić się, że została użyta komenda usuwania, właściwa dla naszej wersji crona.

Teraz, gdy mamy wykłarowaną sytuację, umieścmy coś użytecznego w pliku crontab użytkownika **root**. Większość ludzi zechce uruchamiać updatedb w odstępach tygodniowych, by zapewnić prawidłowe działanie slocate. Ponownie modyfikujemy plik crons.cron, by wyglądał następująco:

Listing 3.14: Plik crontab z życia wzięty

```
22 2 * * 1    /usr/bin/updatedb
```

Powyższe sprawi, że cron będzie uruchamiał updatedb każdego tygodnia, w poniedziałek o 2:22 nad ranem. Powinniśmy teraz dodać plik crontab wykorzystując komendę nowy wpis, zawartą w tabeli powyżej, a następnie ponownie sprawdzić listę zadań.

Listing 3.15: Listowanie zaplanowanych zadań

```
# crontab crons.cron  
# crontab -l
```

Przypuśćmy, że chcemy także dodać emerge --sync do listy zadań wykonywanych codziennie. Możemy to wykonać przez modyfikację pliku crons.cron, a następnie przez użycie odpowiedniej komendy do edycji. Poniższe daje nam możliwość modyfikacji pliku crontab użytkownika w miejscu, bez polegania na zewnętrznych plikach, takich jak crons.cron.

Listing 3.16: Edycja w miejscu pliku crontab

```
# crontab -e
```

Polecenie powinno otworzyć plik crontab użytkownika w edytorze. Chcemy, by emerge --sync uruchamiało się codziennie o 6:30 rano, więc zmieniamy zawartość pliku, by wyglądało to mniej więcej tak:

Listing 3.17: Plik crontab z życia wzięty

```
22 2 * * 1    /usr/bin/updatedb
30 6 * * *    /usr/bin/emerge --sync
(jeśli korzystamy z anacrona, dodajemy następującą linię)
30 7 * * *    /usr/sbin/anacron -s
```

Ponownie sprawdzamy listę zadań zaplanowanych do wykonania, tak jak to czyniliśmy w poprzednim przykładzie. Upewniamy się że zadanie zostało zaplanowane. Jeśli tak jest, to mamy wszystko gotowe.

4. Korzystanie z cronbase

Tak jak wspominaliśmy wcześniej, wszystkie trzy pakiety crona są zależne od sys-process/cronbase. Pakiet cronbase tworzy pliki /etc/cron.{hourly,daily,weekly,monthly} i skrypt o nazwie run-crons. W domyślnym pliku /etc/crontab znajdują się takie wpisy:

Listing 4.1: Domyślny crontab

```
*/15 * * * *      test -x /usr/sbin/run-crons && /usr/sbin/run-crons
0   *   * * *      rm -f /var/spool/cron/lastrun/cron.hourly
0   3   * * *      rm -f /var/spool/cron/lastrun/cron.daily
15  4   * * 6      rm -f /var/spool/cron/lastrun/cron.weekly
30  5   1 * *      rm -f /var/spool/cron/lastrun/cron.monthly
```

Nie będziemy wdawać się w zbytnie szczegóły, opowiemy pokrótko jak uruchamiać poszczególne skrypty co godzinę, co jeden dzień, co tydzień i co miesiąc. Takie ustawienie ma kilka zalet:

- Zostaną uruchomione nawet jeśli komputer był wyłączony w czasie, na który zaplanowano ich wykonanie
- Ułatwia to umieszczanie skryptów przez osoby opiekujące się poszczególnymi pakietami
- Od razu wiadomo gdzie znajdują się wszystkie zadania, ułatwia to tworzenie kopii zapasowych i odtwarzanie części systemu.

Uwaga: Po raz kolejny przypominamy, że Vixie cron automatycznie czyta /etc/crontab, podczas gdy dcron i feron tego nie robią. Więcej szczegółów na ten temat znajduje się w akapacie o crontab dla systemu.

5. Uwagi końcowe

W razie problemów

Jeśli występują problemy z prawidłową pracą crona, to można sprawdzić kolejne punkty z poniższej listy.

- **Czy cron jest uruchomiony?**
Upewniamy się wpisując ps ax | grep cron i oczekując, że cron pojawi się na liście!
- **Czy cron pracuje prawidłowo?**
Próbujemy: * * * * * /bin/echo "foobar" >> /nasz_plik, a następnie upewniamy się że cron działa
- **Czy zadanie uruchamia się poprawnie?**
Próbujemy: * * * * * /bin/foobar > /nasz_plik 2>&1, szukamy błędów w pliku /nasz_plik
- **Czy cron uruchamia zaplanowane zadania?**
Sprawdzamy logi crona w poszukiwaniu błędów, przeważnie jest to plik /var/log/cron.log lub /var/log/messages
- **Czy jest jakiś plik dead.letter?**
Zazwyczaj, gdy wystąpi problem, cron wysyła maila. Sprawdzamy pocztę, a także plik ~/dead.letter

17. Aktualizacja jądra systemu

1. Wprowadzenie

Jądro jest jednym z niewielu rodzajów pakietów w portage, które wymagają interwencji użytkownika do zakończenia procesu aktualizacji. Portage ściągnie i zainstaluje źródła jądra, jednak od tego momentu użytkownik przed osiągnięciem jakichkolwiek zauważalnych zmian musi skompilować jądro samodzielnie.

Chociaż ten przewodnik jest przeznaczony dla użytkowników aktualizujących jądro z jednej wersji do drugiej, może być również użyteczny dla użytkowników przechodzących z jednego rodzaju jądra na inny.

W tym dokumencie jako przykład użycie pakietu gentoo-sources, jednak tych instrukcji można również użyć do innych pakietów, które znajdują się w naszym drzewie.

2. Po co aktualizować jądro systemu?

Generalnie aktualizacje jądra nie przynoszą większych zmian. Jest jednak kilka powodów, dla których je wciąż przeprowadzamy. Jednym z nich jest chęć poznania nowych funkcji lub skorzystania z nowego sterownika, kolejnym jest potrzeba ochrony przed słabymi punktami w bezpieczeństwie lub ostatecznie, posiadanie stabilnego i aktualnego systemu.

Nawet jeśli nie zechcemy aktualizować jądra za każdym razem gdy wyjdzie nowa wersja, zalecane jest robienie tego od czasu do czasu. W przypadku gdy nowe aktualizacje wnoszą poprawki związane z bezpieczeństwem aktualizacja jest bezwzględnie zalecana.

3. Pobieranie nowych źródeł poprzez Portage

Jądro systemu aktualizujemy tak samo jak inne programy - używając emerge. Prawdopodobnie zechcemy to zrobić, gdy pozycja ta pojawi się na liście aktualizacji world. Na przykład:

Listing 3.1: Nowe źródła jądra pojawiły się na liście aktualizacji

```
# emerge -Dup world
Calculating dependencies ...done!
[ebuild   NS ] sys-kernel/gentoo-sources-2.6.9-r2 [2.6.8-r5]
```

Uwaga: Etykieta "NS" w powyższym wyniku polecenia oznacza, że nowe jądro zostanie zainstalowane w nowym slocie. Znaczy to tyle, że źródła starego jądra będą w systemie dopóki nie usuniemy ich ręcznie.

Aktualizację można zainstalować np. tak:

Listing 3.2: Aktualizacja źródeł jądra

```
# emerge -u gentoo-sources
```

Źródła zostaną zainstalowane jako podkatalog w katalogu /usr/src. W powyższym przykładzie będzie to katalog /usr/src/linux-2.6.9-gentoo-r2.

4. Aktualizacja dowiązania symbolicznego /usr/src/linux.

Gentoo wymaga, aby dowiązanie symboliczne /usr/src/linux prowadziło do jądra, którego aktualnie używamy.

Portage może automatycznie zaktualizować dowiązanie symboliczne kiedy zainstalujemy nowe źródła jądra przy użyciu polecenia emerge. Musimy jedynie dodać flagę symlink do zmiennej USE w pliku /etc/make.conf.

Listing 4.1: Przykład zmiennej USE w pliku /etc/make.conf

```
(Dodajemy zmienną symlink)
USE="symlink x86 3dnnow 3dnnowex X aac aalib adns alsa apache2"
```

Do modyfikacji dowiązania symbolicznego możemy również użyć app-admin/eselect.

Listing 4.2: Użycie eselect

```
(Jeżeli nie posiadamy eselect należy go najpierw zainstalować)
# emerge eselect
```

```
(Przeglądanie dostępnych jąder)
# eselect kernel list
Available kernel symlink targets:
 [1]   linux-2.6.9-gentoo-r1
 [2]   linux-2.6.9-gentoo-r2
(Wybieramy odpowiednie jądro)
# eselect kernel set 1
```

Jeśli koniecznie chcemy wykonać powyższą operację ręcznie, poniższy przykład pokazuje jak zrobić dowiązanie do linux-2.6.9-gentoo-r2:

Listing 4.3: Aktualizacja dowiązania symbolicznego/usr/src/linux ręcznie

```
# cd /usr/src
# ln -sfn linux-2.6.9-gentoo-r2 linux
```

5. Konfiguracja, komplikacja i instalacja nowego jądra.

Przy każdej z tych operacji powinniśmy się odnosić do instrukcji, które zawarte są w Podręczniku Gentoo wzorując się na rozdziałach Konfiguracja jądra i Konfiguracja bootloadera. Poniżej napisane jest w skrócie jak to zrobić.

Opcja 1: Automatyczna konfiguracja jądra przy pomocy genkernel

Jeżeli używamy genkernel, musimy jedynie powtórzyć to co robiliśmy gdy pierwszy raz instalowaliśmy jądro systemu.

Po prostu uruchamiamy genkernel w normalny sposób:

Listing 5.1: Uruchamianie genkernel

```
# genkernel all
```

Możemy również użyć dodatkowych parametrów jeżeli zechcemy skonfigurować dodatkowe opcje jądra przy pomocy menuconfig i chcemy by genkernel automatycznie zaaktualizował konfigurację grub, wtedy uruchamiamy genkernel w podany niżej sposób:

Listing 5.2: Uruchamianie genkernel z parametrami

```
# genkernel --menuconfig --bootloader=grub all
```

Więcej informacji znajdziemy w opisie [konfiguracji genkernel](#) lub w [Podręczniku Gentoo](#). Dużą część opcji możemy ustawić w pliku konfiguracyjnym genkernela, czyli /etc/genkernel.conf.

Opcja 2: Konfiguracja ręczna

Aby rozpocząć uruchamiamy menuconfig w katalogu źródeł jądra:

Listing 5.3: Uruchamianie menuconfig

```
# cd /usr/src/linux
# make menuconfig
```

Wybieramy opcje, które są wymagane przez urządzenia i środowisko, w którym będziemy pracować. Po dodatkowe informacje dotyczące konfiguracji jądra, zaglądamy do rozdziału Konfiguracja Jądra w następnym rozdziale

Następnie komplikujemy jądro i kopujemy je na partycję /boot. Ponownie korzystamy z instrukcji zawartych w Podręczniku Gentoo w rozdziale Konfiguracja bootloadera. Jeżeli /boot znajduje się na oddzielnej partycji, upewniamy się iż jest ona zamontowana przed skopiowaniem skompilowanego jądra do niej! Nie zrobienie tego powinno powstrzymać nas od uruchamiania systemu z nowym jądem.

Listing 5.4: Kompilacja i instalacja nowego jądra

```
# make && make modules_install
# mount /boot
```

```
# cp arch/i386/boot/bzImage /boot/bzImage-2.6.9-gentoo-r2
```

Na koniec powinniśmy zaktualizować konfigurację programu ładującego dodając pozycję dotyczącą nowego jądra (nie kasujmy jeszcze starej!) i odmontujemy partycję /boot. Ponownie odnosimy się do Podręcznika Gentoo po dokładniejsze instrukcje opisanego tutaj postępowania.

6. Ponowna instalacja zewnętrznych modułów

Jeśli używamy dodatkowych modułów, które nie są zawarte w źródłach jądra (np. ALSA czy sterowniki graficzne NVIDIA lub ATI), ale znajdują się w portage, muszą być ponownie zainstalowane po każdej aktualizacji jądra systemu. Jest to tak samo proste jak ponowna instalacja innych pakietów. Więcej informacji na ten temat znajdziemy w Podręczniku Gentoo, czytając rozdział o Konfiguracji jądra.

W celu przebudowania wszystkich modułów zainstalowanych za pomocą osobnych ebuildów można skorzystać z prostego programu przygotowanego specjalnie w tym celu. Nosi on nazwę sys-kernel/module-rebuild i reinstaluje moduły dla jądra, którego źródła wskazywane są przez dowiązanie symboliczne /usr/src/linux. Korzystanie z programu jest proste. Po zainstalowaniu go, wystarczy wpisać polecenie module-rebuild populate, aby wypełnić bazę danych zawierającą listę wszystkich zainstalowanych pakietów z dodatkowymi modułami jądra. Następnie trzeba wpisać module-rebuild rebuild, aby program przebudował wszystkie sterowniki.

Więcej informacji na temat programu module-rebuild można uzyskać uruchamiając go bez żadnych parametrów.

7. Uruchamianie systemu z nowym jądrem

Następnie zamykamy wszystkie aplikacje i uruchamiamy ponownie system. Jeśli postępowaliśmy zgodnie z instrukcjami, które są zamieszczone powyżej na liście programu ładującego zobaczymy pozycję odnoszącą się do nowego jądra. Wybieramy ją i uruchamiamy system.

Jeżeli wszystko dobrze skonfigurowaliśmy system uruchomili się z nowym jądrem bez żadnych problemów i będziemy mogli się do niego zalogować oraz rozpocząć pracę. I na tym kończy się proces aktualizacji jądra.

Jeżeli popełniliśmy błąd i system nie wystartował z nowym jądem, wtedy po prostu uruchamiamy go ponownie wybierając w programie ładującym wpis, który odpowiada ostatnio działającej wersji jądra. Następnie przechodzimy z powrotem do Konfiguracji, komplikacji i instalacji nowego jądra - wprowadzamy odpowiednie poprawki, aby naprawić błędy. W niektórych przypadkach (np. gdy po prostu nie zaznaczyliśmy sterownika karty sieciowej, urządzenia audio itp.) nie będziemy musieli nawet uruchamiać ponownie komputera, aby to zrobić.

8. Uruchamianie wielu jąder

Podeczas instalacji nowych źródeł, źródła jądra którego aktualnie używaliśmy nie zostają usunięte. Jest to zamierzone działanie i pozwala w prosty sposób przełączać się, uruchamiając różne jądra.

Przełączanie między różnymi jądrami jest związane z pozostawieniem źródeł jądra w /usr/src oraz pozostawieniem pliku obrazu bzImage w partycji /boot (nawiązując do wpisów w konfiguracji bootloadera). Za każdy razem gdy będziemy uruchamiali system, będziemy proszeni o wybór wersji jądra, która ma być użyta.

9. Usuwanie starego jądra.

Kontynuując poprzedni wątek, po aktualizacji możemy być całkowicie zadowoleni z nowego jądra i nie potrzebować starej wersji. Prostym sposobem na usunięcie wszystkich źródeł z wyjątkiem tych najnowszych jest użycie opcji prune emerge'a. W dalszym ciągu w przykładzie używamy gentoo-sources:

Listing 9.1: Usuwanie starych wersji

```
# emerge -P gentoo-sources
```

W większości przypadków, pliki tymczasowe używane podczas komplikacji będą nadal zapisane w odpowiednim podkatalogu w katalogu /usr/src. Bezpiecznym rozwiążaniem jest usunięcie ich poprzez użycie polecenia rm.

Mogemy również bezpiecznie usunąć moduły, które były używane przez poprzednie jądro. Można to osiągnąć poprzez usunięcie właściwych katalogów w /lib/modules odnoszących się do jądra, które właśnie usuwamy. Należy uważać, aby nie usunąć modułów należących do jądra aktualnie używanego!

Na koniec możemy zamontować partycję /boot i usunąć pliki bzImage. Powinniśmy także wyedytować konfigurację programu ładującego tak, aby nie było już odwołań do tych jąder.

10. Zaawansowane: Użycie starego pliku .config do konfiguracji nowego jądra.

Czasami możliwe jest zaoszczędzenie czasu jeżeli użyjemy ponownie pliku konfiguracyjnego ze starego jądra podczas konfiguracji nowego. Zauważmy jednak, że jest to generalnie niebezpieczne - między kolejnymi wydaniami, wprowadzanych jest zbyt wiele zmian, aby uznać taką metodę za rzetelną.

Jedyną sytuacją kiedy takie postępowanie jest uzasadnione jest przypadek gdy aktualizujemy jądro z jednej wersji Gentoo do innej.

Dla przykładu, różnice pomiędzy gentoo-sources-2.6.9-r1, a gentoo-sources-2.6.9-r2 będą bardzo małe, więc użycie opisywanej metody nie powinno przysporzyć problemów. Jednak nie jest wskazane używanie tej metody w przypadku aktualizacji z wersji np. 2.6.8 do 2.6.9, gdyż różnice pomiędzy oficjalnymi są zbyt duże. Opisana poniżej metoda nie wykazuje tych zmian, co często prowadzi do dodatkowych problemów dla użytkownika poprzez wyłączenie opcji, których wcale nie chciał wyłączać.

Aby ponownie użyć starego pliku .config po prostu kopujemy go w o odpowiednie miejsce i uruchamiamy polecenie make oldconfig. W tym przykładzie, skopujemy ten plik z gentoo-sources-2.6.9-r1 i zainportujemy go do gentoo-sources-2.6.9-r2.

Listing 10.1: Ponowne użycie starej konfiguracji

```
# cd /usr/src/linux-2.6.9-gentoo-r2
# cp ..../linux-2.6.9-gentoo-r1/.config .
# make oldconfig
```

Listing 10.2: Ponowne użycie starej konfiguracji dla genkernela

```
# cd /etc/kernels
# cp kernel-config-x86-2.6.9-gentoo-r1 kernel-config-x86-2.6.9-gentoo-r2
# genkernel all
```

W tym momencie możemy zostać poproszeni o udzielenie odpowiedzi na kilka pytań dotyczących konfiguracji, która zmieniła się pomiędzy tymi dwoma wydaniami. Kiedy to zrobimy, będziemy mogli skompilować i zainstalować jądro w taki sposób w jaki zazwyczaj to czynimy, pomijając jednak proces konfiguracji uruchamiany poleceniem menuconfig.

11. Problemy po aktualizacji jądra?

W szybko rozwijanym jądrze Linuksa nieuniknione jest, że zmiany poczynione między jedną wersją a drugą będą sprawiały jakieś problemy. Jeżeli mamy jakieś rozwiązania pojawiających się problemów z ostatnimi wersjami jąder wspieranych przez Gentoo należy napisać raport o tym rozwiązaniu do nas.

18. Konfiguracja jądra systemu Linuks

1. Wprowadzenie

W Gentoo dostępne są dwie metody konfiguracji jądra. Automatyczna (genkernel) i ręczna. Automatyczna metoda może wydawać się nieco łatwiejsza, ale większość użytkowników i tak wybiera ręczną konfigurację jądra. Powody takiej decyzji mogą być różne, zwykle wymienia się większą elastyczność ręcznej konfiguracji, mniejszy rozmiar jądra, krótszy czas komplikacji, możliwość nauki i nadmier wolnego czasu.

W tym tekście nie ma instrukcji dotyczących metody automatycznej. Jeśli ktoś woli użyć genkernela do komplikacji i instalacji jądra, powinien skorzystać z opisu Konfiguracji jądra genkernel.

Nie opiszemy tu całego procesu konfiguracji od początku do końca. W wielkim stopniu zależy on bowiem od samodzielności użytkownika i jego poziomu wiedzy o używanym sprzęcie. Zamiast tego opiszemy tu najczęściej pojawiające się problemy.

Dokument ten dotyczy instalacji nowszych jąder dla najbardziej popularnych architektur. Niektóre jego fragmenty mogą nie dotyczyć starszych jąder i egzotycznego sprzętu. Większość treści powinna jednak pozostać wspólna.

Zakładamy, że użytkownik posiada źródła jądra Linux na twardym dysku (w katalogu `/usr/src`), że potrafi dostać się do programu konfigurującego jądro za pomocą polecenia `menuconfig` i że potrafi się po nim poruszać. Ten dokument nie jest skierowany do osób, które nie spełniają tych warunków.

- Omówienie źródeł jądra dostępnych w Gentoo to tekst opisujący wszystkie źródła jądra dostępne w drzewie Portage
- Aktualizacja jądra w systemie Gentoo wyjaśnia jak zbudować i zainstalować nowe jądro
- Podręcznik Gentoo zawiera podstawowe informacje na temat konfiguracji jądra.

2. Opis konfiguracji

Podstawy

Cały proces jest stosunkowo prosty. Użytkownikowi przedstawiana jest pewna ilość opcji pogrupowanych w kilkustopniowych menu i musi on wybrać te z nich, które dotyczą jego sprzętu.

Jądro posiada domyślną konfigurację, która jest ustawiana przy pierwszym uruchomieniu `menuconfig` na konkretnym zestawie źródeł. Domyślne opcje są skonfigurowane w bardzo dobry sposób tak, aby były odpowiednie dla większości użytkowników zmuszając ich jedynie do drobnych poprawek konfiguracji. Nie należy wyłączać żadnych domyślnych opcji jeśli nie wie się dokładnie do czego one służą i jakie są konsekwencje ich wyłączenia.

Jeśli jest to pierwsza konfiguracja jądra w życiu, nie należy zmieniać za dużo, a raczej pozostać przy jak największej ilości domyślnych opcji. Należy również być bardzo uważnym, niektóre części konfiguracji są kluczowe dla prawidłowego uruchomienia się systemu.

Wbudowane sterowniki czy moduły

Większość opcji można oznaczyć na trzy sposoby. Jako takie, które wcale nie zostaną zbudowane, takie, które będą wbudowane w jądro na stałe (Y) i takie, które zostaną zbudowane jako moduły (M). Moduły znajdują się w osobnym katalogu na twardym dysku, w odróżnieniu od wbudowanych części jądra, które znajdują się wewnątrz jego obrazu.

Pomiędzy sterownikami wbudowanymi i modularnymi istnieje jedna zasadnicza różnica. Poza kilkoma wyjątkami, jądro nie będzie automatycznie ładowało sterowników zbudowanych jako moduły, niezależnie od tego czy są potrzebne czy nie. Wprawdzie inne składniki systemu mogą zażądać załadowania konkretnego modułu, wciąż jednak zalecamy wbudowywanie wszystkich niezbędnych sterowników bezpośrednio w jądro. Dzięki temu obsługa sprzętu będzie dostępna zawsze, niezależnie od okoliczności.

Niektóre opcje po prostu trzeba wbudować na stałe w jądro. Przykładem takiej opcji jest obsługa używanego na partycjach systemu plików. Jeśli korzysta się z `ext2` i zbuduje się jego obsługę jako moduł, system się po prostu nie uruchomi, ponieważ nie będzie w stanie odczytać modułu `ext2` z partycji, której obsługi nie posiada.

Obsługa sprzętu

Poza wykryciem architektury systemu, program konfiguracyjny nie przeprowadza żadnych testów sprawdzających jaki sprzęt jest zainstalowany w komputerze. Istnieją wprawdzie domyślne opcje konfiguracyjne dla najczęściej używanego sprzętu, ale zwykle nie mają one zastosowania w konkretnym przypadku i konieczne jest ręczne wybranie obsługi używanego sprzętu.

Wymaga to posiadania pewnej wiedzy na temat używanego komputera, znajomości sprzętu jaki jest w nim zainstalowany. Podstawową sprawą jest informacja o tym, jaki chipset ma płyta główna zainstalowana w komputerze, nie jej konkretna nazwa.

Na szczęście istnieją programy, które umożliwiają wykrycie całego zainstalowanego w komputerze sprzętu. Program `lspci` (część pakietu `sys-apps/pciutils`) pozwala zidentyfikować wszystkie karty rozszerzeń podłączone do złącz PCI oraz sprzęt wbudowany w płytę główną, a program `lusb` (z pakietu `sys-apps/usbutils`) pozwala na identyfikację wszystkich urządzeń podłączonych przez porty USB.

Sytuacja jest w pewien sposób skomplikowana przez różne stopnie standaryzacji sprzętu. Jeśli dany komputer nie odbiega za bardzo od domyślnych ustawień, sprzęt po prostu będzie działał. Dyski IDE, czy klawiatura i myszka na PS/2 nie są problemem. Problem pojawi się gdy standardy niemal nie istnieją, tak jak na przykład dla kart sieciowych. W takim wypadku trzeba zidentyfikować konkretny model karty sieciowej i wybrać go z długiej listy podobnych urządzeń w odpowiednim menu programu konfiguracyjnego.

Kilka opcji domyślnych zapewnia tylko podstawową obsługę sprzętu, nie pozwalając na wykorzystanie jego pełnych możliwości. Dotyczy to na przykład obsługi chipsetu IDE, której domyślne ustawienia sprawią, że twarde dyski IDE będą wprawdzie działać, ale ich praca będzie bardzo wolna.

Opcje jądra

Poza obsługą sprzętu, jądro posiada też szereg opcji dotyczących współpracy z różnymi typami oprogramowania. Jednym z przykładów jest obsługa systemów plików, którą trzeba wybrać, aby możliwa była praca na założonych partycjach głównych i dodatkowych (takich jak VFAT czy karty pamięci USB).

Innym przykładem takich możliwości są zaawansowane opcje sieciowe. Aby móc skorzystać z zapór ogniwowych czy routingu, konieczne jest zaznaczenie odpowiednich opcji w konfiguracji jądra.

Gotowi?

Po lekturze tego krótkiego omówienia podstawowych problemów związanych z konfiguracją jądra, należy przystąpić do identyfikacji używanego sprzętu oraz odszukiwania i włączania jego obsługi w programie konfiguracyjnym jądra.

Pozostała część tego tekstu to próba rozwiązywania najczęściej pojawiających się problemów oraz wyjaśnienia kwestii, które najczęściej powodują dezorientację użytkowników. Powodzenia!

3. Najczęstsze problemy

Dyski SATA jako SCSI

Większość nowych komputerów posiada twarde dyski i napędy CD/DVD oparte na Serial ATA częściej niż na starszym IDE.

Obsługa SATA w Linuksie jest implementowana w nakładce o nazwie libata, która znajduje się pod podsystemem SCSI. Z tego powodu sterowniki SATA znajdują się w tej samej sekcji co sterowniki SCSI. Dodatkowo, wszystkie inne urządzenia przechowujące dane są traktowane jako urządzenia SCSI, a zatem potrzebują obsługi dysków/cd-romów SCSI. Dysk twardy SATA ma w systemie nazwę (na przykład) /dev/sda, a dysk SATA CD/DVD /dev/sr0.

Chociaż większość tych sterowników obsługuje kontrolery SATA, libata nie została napisana tylko dla SATA. W niedalekiej przyszłości wszystkie sterowniki IDE również będą przepisane na libata.

Listing 3.1: Konfiguracja libata

```
Device Drivers --->
  SCSI device support --->
    <*> SCSI device support
    <*>   SCSI disk support
    <*>   SCSI CDROM support

  SCSI low-level drivers --->
    <*> Serial ATA (SATA) support
      Z poniżej listy należy wybrać chipset płyty głównej komputera
```

Chipsety IDE i DMA

Pomimo pojawienia się urządzeń SATA, dyski IDE wciąż są bardzo popularne i używane przez bardzo wiele osób. IDE to dość typowy sprzęt, zatem Linux obsługuje niemal wszystkie jego typy bez konieczności wybierania jakichkolwiek dodatkowych opcji.

Niestety, IDE to stara technika i w związku z tym nie jest w stanie obsługiwać szybkich transferów jakie zapewniają nowe urządzenia. Sterownik IDE jest ograniczony do prędkości standardu PIO co nie tylko powoduje, że wymiana danych odbywa się stosunkowo wolno, ale też wywołuje duże obciążenie procesora.

Jeśli komputer został wyprodukowany po 1995 roku, jego kontroler IDE powinien posiadać jeszcze jeden standard przesyłu danych bezpośredniego dostępu do pamięci (Direct Memory Access) nazywany zwykle skrótnicą DMA. Jest on znacznie szybszy i prawie wcale nie zużywa zasobów procesora. Jeśli system działa strasznie wolno i korzysta się z dysku IDE, najbardziej prawdopodobnym źródłem problemu jest właśnie nieużywanie DMA.

Listing 3.2: Sprawdzanie czy DMA jest włączone

```
# hdparm -d /dev/hda

/dev/hda:
using_dma      =  0 (off)
```

Aby włączyć DMA dla urządzeń IDE, wystarczy zaznaczyć w konfiguracji jądra odpowiedni kontroler IDE.

Listing 3.3: Konfiguracja kontrolera IDE

```
Device Drivers --->
ATA/ATAPI/MFM/RLL support --->
<*> ATA/ATAPI/MFM/RLL support
<*> Enhanced IDE/MFM/RLL disk/cdrom/tape/floppy support
[*] PCI IDE chipset support
[*] Generic PCI bus-master DMA support
[*] Use PCI DMA by default when available
Należy wybrać odpowiedni chipset z poniższej listy
```

Kontrolery USB

USB to szeroko stosowane złącze, które umożliwia podłączenie do komputera dużej ilości różnych zewnętrznych urządzeń periferyjnych. Jednym z powodów sukcesu USB jest fakt, że jest bardzo ustandaryzowany, chociaż urządzenia kontrolera (host controller devices) nieco się między sobą różnią. Występują ich trzy typy:

- UHCI (Universal Host Controller Interface) zapewnia obsługę USB 1.1 i jest używane w płytach VIA oraz Intel.
- OHCI (Open Host Controller Interface) zapewnia obsługę USB 1.1 w płytach głównych opartych na chipsetach Nvidia i SiS.
- EHCI (Extended Host Controller Interface) to jeden wspólny kontroler dla USB 2.0, który jest stosowany we wszystkich płytach głównych obsługujących USB 2.0.

Większość komputerów posiada dwa z trzech wymienionych wyżej interfejsów. EHCI (USB 2.0) oraz UHCI lub OHCI (USB 1.1). Ważne jest, aby zaznaczyć w konfiguracji obsługę obu typów kontrolera występujących w komputerze. Wszystkie urządzenia USB 2.0 są kompatybilne z USB 1.1, ale spora część urządzeń USB 1.1 nie działa pod USB 2.0 - bo i po co myszce transfery szybsze niż 1.5mbit na sekundę?

Jeśli nie wybierze się odpowiednich sterowników dla używanych typów USB HCD, porty USB nie będą działały, a podłączone do nich urządzenia nie dostaną zasilania, ani nie będą w stanie się komunikować z komputerem.

Dzięki pewnej sztuczce związanego z lspci (z pakietu sys-apps/pciutils) wykrywanie używanych HCD jest banalnie proste. Ignorując kontroler FireWire, który również spełniał warunki zapytania, odczytujemy, że system używa OHCI i EHCI:

Listing 3.4: Wykrywanie typu HCD za pomocą lspci

```
# lspci -v | grep HCI
00:02.0 USB Controller: nVidia Corporation CK804 USB Controller (rev a2) (prog-if 10
[OHCI])
00:02.1 USB Controller: nVidia Corporation CK804 USB Controller (rev a3) (prog-if 20
[EHCI])
01:0b.0 FireWire (IEEE 1394): Agere Systems FW323 (rev 61) (prog-if 10 [OHCI])
```

Listing 3.5: Konfiguracja USB HCD

```
Device Drivers --->
USB support --->
<*> Support for Host-side USB
--- USB Host Controller Drivers
<*> EHCI HCD (USB 2.0) support
<*> OHCI HCD support
<*> UHCI HCD (most Intel and VIA) support
Należy zaznaczyć obsługę używanych w systemie HCD - można zaznaczyć wszystkie trzy jeśli się nie wie, które wybrać
```

Systemy z wieloma procesorami, hiperwątkowość, dwurdzeniowość

Wiele komputerów posiada więcej niż jeden procesor, ale nie zawsze jest to bardzo oczywiste.

- Wiele procesorów Intela posiada tak zwany hyper-threading, czyli hiperwątkowość, co powoduje, że jeden fizyczny procesor jest traktowany w systemie jak dwa logiczne procesory.
- Niektóre najnowsze procesory Intela i AMD składają się faktycznie z kilku fizycznych procesorów wewnątrz jednego opakowania. Są one nazywane procesorami dual core - czyli dwurdzeniowymi.
- Niektóre najnowocześniejsze komputery posiadają kilka fizycznych procesorów zainstalowanych na jednej specjalnej płycie głównej, co powoduje znaczny wzrost wydajności pracy w stosunku do systemów jednoprocesorowych. Użytkownik zwykle doskonale wie, że posiada taki komputer, gdyż nie są one tanie.

We wszystkich tych przypadkach należy zaznaczyć odpowiednie opcje w jądrze, aby wykorzystać maksimum wydajności oferowanej przez taki sprzęt.

Listing 3.6: Konfiguracja wieloprocesorowości

```
Processor type and features --->
[*] Symmetric multi-processing support
Opcję powyżej należy zaznaczyć w każdym z opisanych powyżej przypadków
[*] SMT (Hyperthreading) scheduler support
Ta opcja dotyczy procesorów z hiperwątkością (HT)
[*] Multi-core scheduler support (NEW)
Ta opcja dotyczy systemów dwurdzeniowych (dual core)
```

x86, obsługa dużej ilości pamięci

W związku z ograniczeniami 32-bitowej przestrzeni adresowej architektury x86, domyślna konfiguracja jądra obsługuje maksymalnie 896MB pamięci RAM. Jeśli w komputerze zainstalowano więcej pamięci, tylko 896MB z niej będzie używane do czasu włączenia opcji "High Memory Support".

Uwaga: To ograniczenie występuje tylko na architekturze x86 (IA32). Inne architektury nie mają problemów z obsługą większych ilości pamięci.

Opcja ta nie jest włączana domyślnie, ponieważ powoduje pewien narzut na wydajność systemu. Nie należy się tym jednak przejmować. Korzyść z posiadania większej ilości pamięci znacznie przewyższą koszt jej obsługi.

Listing 3.7: Obsługa większej ilości pamięci na x86

```
Processor type and features --->
High Memory Support --->
(X) 4GB
( ) 64GB
Opcja 4GB dotyczy wszystkich sytuacji, gdy system posiada mniej niż 4GB pamięci
```

4. Problemy

Zmiany w konfiguracji nie odnoszą skutku

To częsty problem. Użytkownik dokonuje zmian w konfiguracji, ale potem popełnia po drodze jakiś drobny błąd, w rezultacie którego zamiast uruchomić system z jądrem, które właśnie zbudował, uruchamia go z jakiegoś innego obrazu. Wtedy zauważa, że pomimo zmian w konfiguracji jądra ich problem wciąż istnieje i wnioskuje z tego, że zmiany tej konfiguracji nie wpływają na jego problem.

Proces komplikowania i instalowania jądra nie leży w gestii tego dokumentu, można o nim szczegółowo przeczytać w tekście Aktualizacja jądra w systemie Gentoo. W skrócie proces ten polega na konfiguracji jądra, komplikacji, montowaniu partycji /boot, kopiowaniu na nią obrazu jądra i ponownym uruchomieniu komputera. Jeśli któraś z tych czynności zostanie pominięta, zmiany nie odnoszą skutku.

Możliwe jest sprawdzenie czy jądro, z którego uruchomiono system jest tym samym, które znajduje się na dysku, bazując na dacie i godzinie komplikacji.

Zakładając, że używana architektura to x86, a źródła jądra są w katalogu /usr/src/linux:

Listing 5.1: Sprawdzanie jądra

```
# uname -v
#4 SMP PREEMPT Sat Jul 15 08:49:26 BST 2006
Powyższe polecenie wyświetli datę i godzinę skompliowania jądra, z którego uruchomiono system

# ls -l /usr/src/linux/arch/i386/boot/bzImage
-rw-r--r-- 1 dsd users 1504118 Jul 15 08:49 /usr/src/linux/arch/i386/boot/bzImage
Powyższe polecenie wyświetli datę i godzinę skompliowania jądra, które znajduje się na dysku
```

Jeśli czas ten różni się o więcej niż dwie minuty, oznacza to, że podczas instalacji jądra popełniono błąd i że obraz jądra, którego się używa nie jest tym, którego chce się używać.

Moduły nie ładują się automatycznie

Jak już wspominaliśmy wcześniej w tym dokumencie, ustawienie jednej z opcji jądra do zbudowania jako moduł (M) zamiast wbudowania jej na stałe (Y) wiąże się ze sporymi utrudnieniami. Warto powtórzyć to ostrzeżenie, gdyż wielu użytkowników ma przez takie postępowanie problemy.

Gdy sterownik jest wbudowany w jądro, jego kod znajduje się bezpośrednio w jego obrazie (bzImage). Kiedy jądro potrzebuje z niego skorzystać, po prostu go ładuje bez potrzeby jakiekolwiek ingerencji ze strony użytkownika.

Kiedy sterownik budujemy jako moduł, kod znajduje się w zewnętrznym pliku zainstalowanym gdzieś w systemie plików. Kiedy jądro chce z niego skorzystać, nie jest w stanie, ponieważ, z kilkoma wyjątkami, nie robi nic by go najpierw załadować do pamięci. Tę czynność pozostawia użytkownikowi.

Zatem jeśli na przykład obsługa karty sieciowej została zbudowana jako moduł, a następnie okazuje się, że nie działa połączenie sieciowe, najbardziej prawdopodobnym powodem jest to, że moduł ze sterownikiem karty sieciowej nie został załadowany podczas uruchamiania systemu.

Dlatego warto zaoszczędzić sobie tych kilku straconych na naprawy minut i wbudowywać wszystkie sterowniki na stałe w jądro, tak żeby system mógł z nich korzystać za każdym razem gdy są mu potrzebne.

19. Praca z distcc w Gentoo

1. Wprowadzenie

Co to jest distcc?

Distcc to program przeznaczony do dystrybucji komplikacji przez sieć do innych komputerów. Składa się z serwera distccd i programu klienta distcc. Po odpowiednim skonfigurowaniu Distcc może pracować transparentnie wraz z ccache, Portage i Automake.

Używanie distcc do bootstrapu

Jeśli distcc ma pomóc podczas instalacji Gentoo na etapie bootstrapu, należy najpierw przeczytać akapit Używanie distcc do bootstrapu, który znajduje się w dalszej części tego dokumentu.

2. Instalacja

Wymagania

Aby możliwe było skorzystanie z Distcc, wszystkie komputery w sieci muszą posiadać tę samą wersję GCC. Mieszanie kompilatorów w wersjach 3.3.x (gdzie zmienia się x) jest w porządku, jednak mieszanie 3.3.x z 3.2.x **może** przynieść niepożądany rezultat w postaci błędów podczas komplikacji lub uruchomienia.

Instalacja distcc

Jest kilka rzeczy z którymi należy się zapoznać przed rozpoczęciem instalacji distcc.

Distcc dostarczane jest wraz z graficznym monitorem do nadzorowania zadań, które komputer rozsyła do komplikacji. Jeśli używane jest środowisko Gnome, należy dodać flagę 'gnome' do zmiennej USE. Jeśli Gnome nie jest używany, nadal można używać graficznego monitora poprzez dodanie flagi 'gtk'.

Listing 2.1: Instalacja distcc

```
# emerge distcc
```

Ważne: Distcc musi zostać zainstalowane na wszystkich komputerach biorących udział w budowaniu programów.

Konfiguracja Portage do współpracy z distcc

Konfiguracja Portage tak, aby korzystał z distcc jest łatwa. Należy wykonać następujące czynności na każdym systemie, który będzie brał udział w dystrybucji komplikacji:

Listing 2.2: Integracja distcc z Portage

```
# emerge distcc
# nano -w /etc/make.conf
(Należy podmienić N liczbą odpowiednią dla danej konfiguracji)
(Najczęściej N to podwojona liczba dostępnych procesorów + 1)
MAKEOPTS="-jN"
(Trzeba dodać distcc do zmiennej FEATURES)
FEATURES="distcc"
```

Wybieranie komputerów do pracy distcc

Należy użyć polecenia distcc-config, aby ustawić listę komputerów. Poniżej przykład kilku wpisów które mogłyby się znaleźć na takiej liście:

Listing 2.3: Przykład definicji komputerów

192.168.0.1	192.168.0.2	192.168.0.3
192.168.0.1/2	192.168.0.2	192.168.0.3/10
192.168.0.1:4000/2	192.168.0.2/1	192.168.0.3:3632/4
@192.168.0.1	@192.168.0.2:/usr/bin/distccd	192.168.0.3

(Istnieje także kilka innych metod ustawienia komputerów. Więcej informacji można znaleźć w podręczniku - man distcc)

Jeśli zamierzamy kompilować także na lokalnej maszynie należy dodać do

tej listy wpis 'localhost'. Oczywiście jeśli nie chcemy kompilować na lokalnej maszynie (co jest bardzo częste) należy pominąć ten wpis. Na wolnych maszynach użycie wpisu localhost może tylko spowolnić proces komplikacji. Należy przetestować swoje ustawienia pod kątem wydajności

To może wyglądać skomplikowanie, ale w większości przypadków zadziała wariant z linii 1 lub 2.

Jako że większość ludzi nie będzie używać linii 3 i 4, po więcej informacji odwołamy się do dokumentacji distcc (man distcc).

Ustawienie przykładu z pierwszej linii:

Listing 2.4: Przykładowe polecenie ustalające listę komputerów

```
# /usr/bin/distcc-config --set-hosts "192.168.0.1 192.168.0.2 192.168.0.3"
```

Należy dostosować do konkretnych potrzeb plik /etc/conf.d/distccd upewniając się, że użyta jest dyrektywa --allow, co umożliwia dostęp tylko zaufanym komputerom. Dla zwiększenia bezpieczeństwa, należy także użyć dyrektywy --listen, aby powiedzieć demonowi distcc na jakim adresie IP ma nasłuchiwać (dla systemów z wieloma adresami IP). Więcej informacji o bezpieczeństwie w distcc można znaleźć w zarysie bezpieczeństwa w distcc.

Ważne: Użycie --allow i --listen jest kluczowe. Aby uzyskać więcej informacji, należy przeczytać dokumentację man programu distcc lub powyższy dokument o bezpieczeństwie.

Teraz trzeba uruchomić demona distcc na wszystkich komputerach:

Listing 2.5: Uruchamianie demona distcc

```
(Uruchamianie distcc start systemu)
# rc-update add distccd default
(Uruchomienie demona distcc)
# /etc/init.d/distccd start
```

Konfiguracja distcc do pracy z automake

W niektórych przypadkach jest to łatwiejsze niż konfiguracja z Portage. Jedyne co trzeba zrobić to zmiana zmiennej PATH tak, aby zawierała ścieżkę /usr/lib/distcc/bin przed katalogiem zawierającym gcc (/usr/bin). Jednakże jest pewien wyjątek. Jeśli używane jest ccache należy dodać distcc zaraz po ccache:

Listing 2.6: Ustawianie zmiennej path

```
# export PATH="/usr/lib/ccache/bin:/usr/lib/distcc/bin:${PATH}"
Można to dodać do swojego pliku .bashrc lub jego odpowiednika by mieć
ustawianą zmienną PATH przy każdym logowaniu
```

Następnie zamiast normalnego wywołania make, należy wpisać make -jN (gdzie N jest liczbą całkowitą). Wartość liczby N zależy od konfiguracji sieci oraz rodzaju komputerów użytych do komplikacji. Trzeba przetestować różne ustawienia by znaleźć te najbardziej optymalne pod względem wydajności.

3. Kompilacja skrośna

Kompilacja skrośna to używanie jednej architektury do komplikacji programów dla innej. Może być to np. użycie Athlona (i686) do komplikacji programu na K6-2 (i586) albo użycie SPARC-a do komplikacji programu dla PPC (Power PC). Na temat komplikacji skrośnej można poczytać w Przewodniku komplikacji skrośnej za pomocą DistCC.

4. Używanie distcc do bootstrapu

Krok 1: konfiguracja Portage

Należy uruchomić komputer z nowym Gentoo LiveCD i postępować zgodnie z Podręcznikiem instalacji Gentoo aż do części dotyczącej bootstrapu (Więcej informacji na temat bootstrapu można znaleźć w odpowiedziach na najczęściej zadawane pytania). Następnie trzeba skonfigurować Portage do pracy z distcc:

Listing 4.1: Wstępna konfiguracja

```
# nano -w /etc/make.conf
(Dodanie distcc do zmiennej FEATURES
FEATURES="distcc"
(Modyfikacja zmiennej MAKEOPTS by zawierała -jN, gdzie N to 2 razy
liczba dostępnych procesorów +1)
MAKEOPTS="-jN"
```

Listing 4.2: Ustawienie zmiennej path

```
# export PATH="/usr/lib/ccache/bin:/usr/lib/distcc/bin:${PATH}"
```

Krok 2: instalacja distcc

Przed instalacją distcc, do pliku /etc/passwd powinien zostać dodany użytkownik o nazwie distcc:

Listing 4.3: Tworzenie użytkownika distcc

```
# echo "distcc:x:240:2:distccd:/dev/null:/bin/false" >>/etc/passwd
```

Ważne: Dodawanie użytkowników w ten sposób jest bardzo niewłaściwe. Robimy to tylko dla tego, ponieważ na tym etapie instalacji nie mamy dostępu do narzędzia useradd (które jest normalnie używane do dodawania kont).

Instalacja distcc:

Listing 4.4: Instalacja distcc w nowym systemie

```
# USE='-*' emerge --nodeps sys-devel/distcc
```

Krok 3: Konfiguracja distcc

Trzeba uruchomić polecenie distcc-config --install:

Listing 4.5: Końcowa konfiguracja distcc

(Należy podmienić host1, host2, ... adresami IP odpowiednich komputerów)

```
# /usr/bin/distcc-config --set-hosts "localhost host1 host2 host3 ..."
```

Na przykład: /usr/bin/distcc-config --set-hosts "localhost
192.168.0.4 192.168.0.6"

Teraz distcc jest skonfigurowane do bootstrapu! Należy kontynuować oficjalną instrukcję instalacji i nie zapomnieć ponownie emergować distcc zaraz po emerge system. Zapewni to poprawną instalację wszystkich pakietów wymaganych przez distcc.

Uwaga: Podczas bootstrapu i emerge system może się wydawać że distcc nie jest używane. Jest to pożąданie zachowanie jako że niektóre ebuildy nie współpracują z distcc, więc go wyłączają.

5. Rozwiązywanie błędów

Nie wszystkie pakiety korzystają z distcc

Podczas instalacji różnych pakietów można zauważać że niektóre nie są rozsyłane (i nie są budowane wspólnie). Dzieje się tak, ponieważ niektóre pliki Makefile programów po prostu nie potrafią współpracować z równoległyimi komplikacjami i twórcy ebuildów do tych programów muszą wyłączać tę możliwość.

Czasem distcc może spowodować, że pakiet nie chce się skompilować. Jeśli tak się zdarzy, należy to [zgłosić](#).

Mieszanie wersji GCC

Jeśli na naszych komputerach zainstalowane są różne wersje GCC, prawdopodobnie będzie wiele dziwnych problemów. Rozwiążaniem jest zapewnienie że wszystkie systemy będą miały tą samą wersję GCC.

Ostatnie aktualizacje Portage sprawiły, że używa on \${CHOST}-gcc zamiast gcc. Oznacza to że jeśli używamy maszyn i686 z innymi typami (i386, i586) szybko wpadniemy w kłopoty. Rozwiążaniem tego może być wykonanie polecenia export CC='gcc' CXX='c++' lub wpisanie go na stałe do /etc/make.conf.

Ważne: Wykonanie powyższego polecenia zmienia zachowanie Portage i może nieść za sobą dziwne rezultaty w przyszłości. Należy robić to tylko wtedy, gdy mieszka się różne ustawienia CHOST.

6. Dodatki do distcc

Monitory distcc

Distcc dostarczane jest z dwoma monitorami. Pierwszy, działający w trybie tekstowym jest instalowany zawsze i nazywa się distccmon-text. Pierwsze jego uruchomienie może wprowadzić w zakłopotanie, ale jest on naprawdę całkiem łatwy w użyciu. Jeśli zostanie uruchomiony bez żadnych parametrów uruchomi się raz. Jednakże jeśli poda mu się jako argument liczbę, będzie odświeżał ekran co określona liczbą sekund.

Kolejny program monitorujący jest instalowany tylko wtedy gdy mamy ustawioną w zmiennej USE którąkolwiek z flag gtk lub gnome. Oparty jest on na GTK+, działa w środowisku X i ma bardzo ładny wygląd. W Gentoo ten monitor nazwany został distccmon-gui. W innych źródłach może być nazywany distccmon-gnome.

Listing 6.1: Uruchamianie monitorów

```
# distccmon-text N
(lub)
# distccmon-gui
Aby nadzorować zachowanie distcc z Portage można użyć:
# DISTCC_DIR="/var/tmp/portage/.distcc/" distccmon-text N
(lub)
# DISTCC_DIR="/var/tmp/portage/.distcc/" distccmon-gui
```

Ważne: Jeśli katalog distcc ma inną lokalizację, należy odpowiednio ustawić zmienną DISTCC_DIR.

20. Konfiguracja routera

1. Wprowadzenie

Budowa własnego routera ze starych niepotrzebnych części to spora oszczędność w porównaniu z kupnem gotowego rozwiązania, np. firmy Linksys. Największą zaletą takiej decyzji jest niemal całkowita kontrola nad połączeniem. Pozostałe możliwości są zależne tylko od naszej wyobraźni.

W tym tekście omówimy konfigurowanie translacji adresów (NAT) na routerze (jądro oraz iptables), dodawanie i konfigurację podstawowych usług (serwer nazw (DNS) przez dnsmasq, dhcp przez dhcpcd, ADSL przez rp-pppoe), a zakończymy na bardziej wyszukanych i dających więcej satysfakcji usługach jak przekazywanie portów, kształtowanie ruchu, serwery proxy/cachowanie, itp.

Jest kilka podstawowych wymagań jakie trzeba poznać przed rozpoczęciem pracy. Po pierwsze potrzebny będzie komputer z co najmniej dwiema kartami sieciowymi. Do tego przyda się znajomość ustawień połączenia sieciowego (IP, DNS, adres bramy, nazwa użytkownika i hasło). Niezbędna jest również odrobina wolnego czasu i zamiłowanie do Gentoo.

Konwencje użyte w tekście:

- eth0 - NIC podłączona do sieci lokalnej (LAN)
- eth1 - NIC podłączona do sieci rozległej (WAN)
- LAN korzysta z prywatnych adresów 192.168.0.xxx
- Router ma przypisany adres 192.168.0.1
- Router pracuje na jądrze Linux 2.4 lub 2.6. Jąder 2.0 i 2.2 nie wspieramy.

Ważne: W związku z zachowaniem środków ostrożności, zalecane jest wyłączenie wszystkich niepotrzebnych usług na routerze do czasu uruchomienia firewalla. Aby przejrzeć aktualnie uruchomione usługi, należy wykonać polecenie rc-status.

2. Konfiguracja jądra

Jądro powinno posiadać sterowniki dla wszystkich używanych kart sieciowych. W celu sprawdzenia czy odpowiednie sterowniki są włączone należy wykonać polecenie ifconfig. Jego wynik może nieco różnić się od tego podanego poniżej, nie jest to jednak wielkim problemem. Najważniejsze, że wyświetla się interfejsy.

Listing 2.1: Sprawdzanie kart sieciowych

```
# ifconfig -a
eth0      Link encap:Ethernet HWaddr 00:60:F5:07:07:B8
          BROADCAST MULTICAST MTU:1500 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
          Interrupt:11 Base address:0x9800

eth1      Link encap:Ethernet HWaddr 00:60:F5:07:07:B9
          BROADCAST MULTICAST MTU:1500 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
          Interrupt:10 Base address:0x9400
```

Jeśli nie pojawią się interfejsy obu kart sieciowych i nie jest się pewnym jakie to są karty, warto uruchomić program lspci | grep Ethernet (instalowany polecienniem emerge pciutils). Potem należy przejść do konfiguracji jądra i dodać obsługę odpowiedniego sterownika.

Kolejne niezbędne moduły to obsługa iptables i NAT oraz - opcjonalnie - kształtowania ruchu. Na liście poniżej znajdują się opcje wymagane (*), zalecane (x) oraz dotyczące kształtowania ruchu (s). Bez znaczenia jest fakt, czy wbudujemy je na stałe, czy też zbudujemy w postaci modułów. Jeśli jakiś moduł będzie potrzebny to powinien zostać załadowany automatycznie. Sprawdzanie czy wszystko co potrzebne się ładuje pozostawiamy jako rozrywkę dla czytelnika.

Listing 2.2: Opcje sieciowe

```
Networking options  ---
[*] TCP/IP networking
```

```

[*] IP: advanced router
[*] Network packet filtering (replaces ipchains)
Używając jądra 2.4.x, musimy zaznaczyć poniższą opcję dla DHCP:
[*] Socket Filtering

IP: Netfilter Configuration --->
[*] Connection tracking (required for masq/NAT)
[x] FTP protocol support
[x] IRC protocol support
[*] IP tables support (required for filtering/masq/NAT)
[*] IP range match support
[x] MAC address match support
[*] Multiple port match support
[*] Packet filtering
[*] REJECT target support
[x] REDIRECT target support
[*] Full NAT
[*] MASQUERADE target support
[s] Packet mangling
[s] MARK target support
[x] LOG target support

QoS and/or fair queueing --->
[s] QoS and/or fair queueing
[s] HTB packet scheduler
[s] Ingress Qdisc

[a] PPP (point-to-point protocol) support
[a] PPP filtering
[a] PPP support for async serial ports
[a] PPP support for sync tty ports
[a] PPP Deflate compression
[a] PPP BSD-Compress compression
[a] PPP over Ethernet

```

Uwaga: Pomiędzy jądrami serii 2.4 i 2.6 mogą pojawić się nieznaczne różnice, wierzymy jednak, że czytelnicy poradzą sobie z nimi wszystkimi. Nawet pomiędzy poszczególnymi wydaniami jądra 2.6 niektóre z tych opcji są przemieszczone. Powodzenia!

3. WAN (czyli Internet)

Wstęp

Istnieje wiele różnych sposobów na łączenie się z Internetem, zajmiemy się jednak tylko tym, na którym znam się najlepiej, czyli ADSL (PPPoE). Jeśli ktoś posiada rozsądny opis związanego z innymi typami połączeń może mi go podesłać. Niektóre akapity mogą nie dotyczyć określonej sytuacji, należy je wtedy po prostu pominąć. W tym rozdziale opowiem o podłączaniu routera do Internetu poprzez interfejs eth1.

ADSL i PPPoE

Cale oprogramowanie dla PPPoE znajduje się w jednym małym pakiecie - [Roaring Penguin](#). Jego instalacja w Gentoo odbywa się za pomocą polecenia emerge rp-pppoe. Na początku dokumentu wspominaliśmy o nazwie użytkownika i haśle, należy je umieścić w pliku /etc/ppp/pppoe.conf.

Uwaga: Aby poniższe instrukcje zadziałyły potrzebny będzie pakiet baselayout w wersji 1.11.14 lub nowszy.

Listing 3.1: Konfiguracja interfejsu eth1

(Zamieniamy 'vla9h924' na odpowiednią nazwę użytkownika, a 'password' na odpowiednie hasło)

```
# nano /etc/ppp/pap-secrets
# Kolejno: Klient, serwer, tajemnica:
```

```
"vla9h924" * "password"
# nano /etc/conf.d/net
# Wpis przypisujący połaczenie adsl do eth1:
ifconfig_eth1=( "adsl" )
user_eth1=( "vla9h924" )
# ln -s net.lo /etc/init.d/net.eth1
# rc-update add net.eth1 default
# /etc/init.d/net.eth1 start
```

Ostrzeżenie: Przy podnoszeniu interfejsu DSL zostanie utworzone ppp0. Jest to dość mylące, gdyż karta sieciowa tego interfejsu nazywa się eth1. Od tej pory w przykładach utożsamiamy eth1 z ppp0.

Modem kablowy i dynamiczne/statyczne IP

Posiadacze statycznych adresów IP potrzebują paru informacji więcej niż użytkownicy adresów przydzielanych dynamicznie. Będą im potrzebne następujące dane: adres IP, adres bramki i adresy serwerów DNS.

Listing 3.2: Konfiguracja interfejsu eth1

Użytkownicy posiadający adres IP przydzielany dynamicznie:

```
# emerge dhcpcd
# nano /etc/conf.d/net
Potrzebny będzie wpis:
ifconfig_eth1=( "dhcp" )
```

Użytkownicy posiadający statyczny adres IP:

```
# nano /etc/conf.d/net
Potrzebny będzie wpis:
config_eth1=( "66.92.78.102 broadcast 66.92.78.255 netmask 255.255.255.0" )
routes_eth1=( "default gw 66.92.78.1" )
# nano /etc/resolv.conf
Dodajemy jeden serwer na linię:
nameserver 123.123.123.123
```

Polecenia dla wszystkich:

```
# ln -s net.lo /etc/init.d/net.eth1
# rc-update add net.eth1 default
# /etc/init.d/net.eth1 start
```

4. LAN

Czynności opisane w tym akapicie w porównaniu z tymi z poprzedniej części są banalnie proste do przeprowadzenia.

Listing 4.1: Konfiguracja interfejsu eth0

```
# nano /etc/conf.d/net
Dodajemy tam:
config_eth0=( "192.168.0.1 broadcast 192.168.0.255 netmask 255.255.255.0" )
# rc-update add net.eth0 default
# /etc/init.d/net.eth0 start
```

5. Usługi w LAN-ie

Serwer DHCP

Byłoby miło gdyby użytkownicy sieci mieli wszystko automatycznie konfigurowane zaraz po włączeniu komputera bez potrzeby jakiekolwiek ingerencji z ich strony. Nie musieliby pamiętać tych wszystkich liczb, ani gubić się w zakręconych plikach konfiguracyjnych. Życie byłoby wtedy znacznie prostsze i piękniejsze. I właśnie taki stan powszechnej szczęśliwości może nam zapewnić serwer DHCP.

DHCP to protokół umożliwiający dynamiczną i w pełni automatyczną konfigurację innych komputerów. Wystarczy uruchomić na routerze serwer DHCP (dhcpcd), podać mu wszystkie informacje o potencjalnych klientach (adresy IP, adresy serwerów DNS, bram sieciowych itp.), a komputer podłączony do serwera zostanie dobrze skonfigurowany już przy starcie systemu. Więcej informacji o DHCP znajduje się w Wikipedii.

Skorzystamy z pakietu o nazwie dnsmasq, który zawiera w sobie usługi DHCP i DNS. Teraz skupimy się na usłudze DHCP. Informacje o tym jak uruchomić inny serwer DHCP znajdują się w akapicie zatytułowanym "Zabawki". Lektura komentarzy z pliku /etc/dnsmasq.conf może pomóc w majstrowaniu przy serwerze DHCP, aby lepiej go dostosować do konkretnych potrzeb. Domyślana konfiguracja powinna zadowolić większość osób.

Listing 5.1: Konfiguracja dhcpcd

```
# emerge dnsmasq
# nano /etc/dnsmasq.conf
Wystarczy dodanie jednej linii i DHCP będzie działać:
dhcp-range=192.168.0.100,192.168.0.250,72h
Udostępniamy dnsmasq tylko dla naszego LAN-u
interface=eth0
# rc-update add dnsmasq default
# /etc/init.d/dnsmasq start
```

I tak nasz mały router stał się najprawdziwszym serwerem DHCP! Teraz wystarczy podłączyć komputery do sieci i sprawdzić, czy wszystko działa jak należy. Jeśli chodzi o systemy Windows to musimy przejść do właściwości protokołu TCP/IP i zaznaczyć tam opcję 'Pobierz adresy serwerów DNS automatycznie'. Czasami zmiany nie są wczytywane automatycznie i natychmiastowo, dlatego konieczne może okazać się przejście do wiersza poleceń i wpisanie tam polecień ipconfig /release oraz ipconfig /renew. To tyle o Windowsie - wracamy do Gentoo.

Serwer DNS

Ludzie nie lubią zapamiętywać liczb, znacznie lepiej przyswajają zwykłe nazwy, dlatego łatwiej zapamiętać adres ebay.com niż 66.135.192.8. W związku z tym istnieje zapotrzebowanie na coś takiego jak serwer DNS. Serwery DNS znajdują się w całym Internecie, a ich zadaniem jest zwracanie ciągu '66.135.192.87' za każdym razem gdy ktoś będzie chciał odwiedzić stronę ebay.com. Więcej informacji o DNS można znaleźć w Wikipedii.

W związku z tym, że korzystamy z dnsmasq jako serwera DHCP i zawiera on w sobie również DNS, nic nie musimy tu ustawać. Nasz router już zapewnia obie usługi wszystkim połączonym z nim komputerom.

Oczywiście można wybrać inny serwer DNS jeśli ma się taką potrzebę.

NAT (maskarada IP)

W tym momencie użytkownicy naszej sieci mogą komunikować się między sobą oraz przeszukiwać bazę domen, ale wciąż tak naprawdę nie mogą swobodnie łączyć się z Internetem. Administrator może być z tego powodu zadowolony (bo ma nieobciążone łącze), ale użytkownicy na pewno zbyt długo takiego stanu rzeczy nie będą znośić.

W celu umożliwienia użytkownikom prywatnego LAN-u łączenia się z Internetem musimy skonfigurować NAT. Rozwiążanie to przydaje się zwłaszcza wtedy, gdy mamy do dyspozycji tylko jeden zewnętrzny adres IP (co jest bardzo częstym przypadkiem u wielu ISP), a chcemy podłączyć do Internetu komputery w całym domu. Więcej informacji o NAT można znaleźć w Wikipedii.

Uwaga: Zanim zaczniemy, należy się upewnić, że mamy zainstalowany pakiet iptables. W większości systemów jest instalowany automatycznie, ale może również się zdarzyć, że go brakuje. Jeśli go nie ma to instalujemy go za pomocą polecenia emerge iptables.

Listing 5.2: Konfiguracja iptables

Na początek czyścimy regułki:

```
# iptables -F
# iptables -t nat -F
```

Domyślne zasady dla pakietów nie pasujących do żadnej regułki

```
# iptables -P INPUT ACCEPT
# iptables -P OUTPUT ACCEPT
# iptables -P FORWARD DROP
```

Przykłady można przekleić

```
# export LAN=eth0
# export WAN=eth1
```

Potem blokujemy usługi tak, aby były dostępne tylko dla sieci LAN

```
# iptables -I INPUT 1 -i ${LAN} -j ACCEPT
# iptables -I INPUT 1 -i lo -j ACCEPT
```

```
# iptables -A INPUT -p UDP --dport bootps -i ! ${LAN} -j REJECT
# iptables -A INPUT -p UDP --dport domain -i ! ${LAN} -j REJECT

(Opcjonalnie) Pozwalamy na dostęp do naszego serwera SSH z Internetu
# iptables -A INPUT -p TCP --dport ssh -i ${WAN} -j ACCEPT

Upuszczamy pakiety TCP/UDP dla uprzywilejowanych portów
# iptables -A INPUT -p TCP -i ! ${LAN} -d 0/0 --dport 0:1023 -j DROP
# iptables -A INPUT -p UDP -i ! ${LAN} -d 0/0 --dport 0:1023 -j DROP

Ostatecznie dodajemy regułki dla NAT
# iptables -I FORWARD -i ${LAN} -d 192.168.0.0/255.255.0.0 -j DROP
# iptables -A FORWARD -i ${LAN} -s 192.168.0.0/255.255.0.0 -j ACCEPT
# iptables -A FORWARD -i ${WAN} -d 192.168.0.0/255.255.0.0 -j ACCEPT
# iptables -t nat -A POSTROUTING -o ${WAN} -j MASQUERADE

Informujemy jądro o chęci przekazywania IP
# echo 1 > /proc/sys/net/ipv4/ip_forward
# for f in /proc/sys/net/ipv4/conf/*/_rp_filter ; do echo 1 > $f ; done

Poniższe polecenia sprawia, że przy każdym uruchomieniu routera
nie będziemy musieli wpisywać ręcznie wszystkich regułek
# /etc/init.d/iptables save
# rc-update add iptables default
# nano /etc/sysctl.conf

Dodajemy przez odkomentowanie następujące linijki:
net.ipv4.ip_forward = 1
net.ipv4.conf.default.rp_filter = 1
Osoby bez stałego dostępu do Internetu zapewnia zechcą włączyć tę opcję:
net.ipv4.ip_dynaddr = 1
```

Po wykonaniu powyższych poleceń komputery z naszej małej sieci powinny mieć dostęp do Internetu równie swobodny, jak gdyby były podłączone do niego bezpośrednio.

Opcja ip_dynaddr jest potrzebna do wyboru systemu dial on demand, gdy provider nie zapewnia stałego adresu. Pozwala to na obejście problemu gdy połączenie jest inicjowane przed pełnym skonfigurowaniem interfejsu. Pozwala to na ułatwienie życia osobom korzystającym z sieci za pomocą routera.

6. Zabawki (coś na długie deszczowe dni)

Wstęp

Wierzycie czy nie - to już wszystko. Teraz zajmiemy się dodatkowymi gadżetami, które na pewno zainteresują wiele osób. Wszystkie są całkowicie opcjonalne.

Przekazywanie portów.

Czasem zachodzi potrzeba uruchomienia jakichś usług na komputerze znajdującym się za routera lub też umożliwienie zdalnego do niego połączenia. Może chodzić np. o uruchomienie na jednym lub kilku komputerach za NAT-em serwera FTP, HTTP, SSH lub VNC. Jedynym ograniczeniem wtedy jest to, że można mieć tylko jedną taką usługę na każdym porcie naszego routera. Na przykład nie ma możliwości przekazywania połączeń na trzy serwery FTP za routera i łączenia się przez nie poprzez port 21. Jeśli jednak to konieczne, zawsze można uruchomić pozostałe na innych portach - np. 123 czy 567.

Każda regułka dla przekazywania portów ma postać iptables -t nat -A PREROUTING [-p protokół] --dport [zewnętrzny port na routerze] -i \${WAN} -j DNAT -to [ip/port na które chcemy przekazywać]. Iptables nie akceptuje nazw stacji roboczych, nie trzeba też podawać portu docelowego na wewnętrznym komputerze. Zawsze jest to ten sam port co na komputerze przekierowującym. Więcej informacji na ten temat znajduje się w man iptables.

Listing 6.1: Przekazywanie portów

```
Przykłady można przekleić
# export LAN=eth0
# export WAN=eth1

Przekazywanie portu 2 dla ssh na wewnętrznej stacji roboczej
# iptables -t nat -A PREROUTING -p tcp --dport 2 -i ${WAN} -j DNAT --to 192.168.0.2:22

Przekazywanie FTP na wewnętrzną stację roboczą
# iptables -t nat -A PREROUTING -p tcp --dport 21 -i ${WAN} -j DNAT --to 192.168.0.56
```

```

Przekazywanie HTTP na wewnętrzną stację roboczą
# iptables -t nat -A PREROUTING -p tcp --dport 80 -i ${WAN} -j DNAT --to 192.168.0.56

Przekazywanie VNC na wewnętrzną stację roboczą
# iptables -t nat -I PREROUTING -p tcp --dport 5900 -i ${WAN} -j DNAT --to 192.168.0.2
# iptables -t nat -I PREROUTING -p tcp --dport 5901 -i ${WAN} -j DNAT --to
192.168.0.3:5900
Aby mieć VNC na 192.168.0.3, należy po prostu dodać ':1' do nazwy routera

Przekazywanie BitTorrent
# iptables -t nat -A PREROUTING -p tcp --dport 6881:6889 -i ${WAN} -j DNAT --to
192.168.0.2

Przekazywanie eDonkey/eMule
# iptables -t nat -A PREROUTING -p tcp --dport 4662 -i ${WAN} -j DNAT --to 192.168.0.55

Wsparcie dla Warp Pipe z Game Cube
# iptables -t nat -A PREROUTING -p udp --dport 4000 -i ${WAN} -j DNAT --to 192.168.0.56

Interaktywna pomoc techniczna Playstation2
# iptables -t nat -A PREROUTING -p tcp --dport 10070:10080 -i ${WAN} -j DNAT --to
192.168.0.11
# iptables -t nat -A PREROUTING -p udp --dport 10070:10080 -i ${WAN} -j DNAT --to
192.168.0.11

Xbox Live
# iptables -t nat -A PREROUTING -p tcp --dport 3074 -i ${WAN} -j
# DNAT --to 192.168.0.69
# iptables -t nat -A PREROUTING -p udp --dport 3074 -i
# ${WAN} -j DNAT --to 192.168.0.69
# iptables -t nat -A PREROUTING -p udp --dport 88
# -i ${WAN} -j DNAT --to 192.168.0.69

```

Identd (dla IRC)

Usługa ident jest bardzo silnie wykorzystywana przez IRC. Teraz, gdy użytkownicy sieci znaleźli się za routerem należy zapewnić identyfikację zarówno im jak i naszemu routerowi. Jednym z serwerów stworzonych w tym celu jest midentd.

Listing 6.2: Konfiguracja ident

```

# emerge midentd
# rc-update add midentd default
# /etc/init.d/midentd start

```

W Portage znajduje się jeszcze kilka innych serwerów tego typu. Zależnie od potrzeb warto zwrócić uwagę na oidentd lub fakeidentd.

Serwer czasu

Prawidłowe ustawienie czasu na serwerze również może dać nam wiele korzyści. Jednym ze sposobów osiągnięcia tego może być sieciowy protokół czasu (NTP), z którego można zacząć korzystać po zainstalowaniu pakietu ntp, zawierającego zarówno serwer jak i program kliencki.

Większość ludzi uruchamia na swoich komputerach tylko program kliencki ntp. Oczywiście, im więcej klientów na świecie, tym większe obciążenie serwery ntp muszą wytrzymać. W warunkach domowych możemy trochę odciążyć publiczne serwery, samodzielnie dostarczając aktualny czas do wszystkich naszych komputerów. Dodatkową korzyścią może być również to, że aktualizacje czasu będą w tym przypadku znacznie szybsze. Wszystko co musimy zrobić to uruchomić na naszym routerze serwer czasu, który będzie się synchronizował z publicznymi serwerami w Internecie, jednocześnie dostarczając poprawny czas do reszty naszych komputerów. Pierwsze polecenie jakie należy wykonać to emerge ntp na routerze.

Listing 6.3: Konfigurowanie serwera NTP

```
# nano /etc/conf.d/ntp-client
Domyślne ustawienia powinny być dobre, ale można również dostosować je do indywidualnych
potrzeb
# rc-update add ntp-client default

# nano /etc/ntp.conf
Dodajemy następujące linie:
restrict default ignore
restrict 192.168.0.0 mask 255.255.255.0 notrust nomodify notrap
To pozwoli na korzystanie z Twojego serwera ntp tylko klientom, którzy posiadają
adres IP z zakresu 192.168.0.xxx
# nano /etc/conf.d/ntpd
Domyślne ustawienia powinny być dobre, ale można również dostosować je do indywidualnych
potrzeb:
# rc-update add ntpd default

# /etc/init.d/ntp-client start
# /etc/init.d/ntpd start
```

Uwaga: Przy konfiguracji serwera należy się upewnić, że jest otwarta możliwość komunikacji do i z portu ntp (123/udp). Klient wymaga jedynie możliwości łączenia się z portu 123 udp.

Następnie na wszystkich komputerach należących do sieci należy wykonać polecenie emerge ntp. Ich konfiguracja będzie znacznie prostsza.

Listing 6.4: Konfiguracja klienta NTP

```
# nano /etc/conf.d/ntp-client
Należy dokonać zmiany NTPCLIENT_OPTS
z 'pool.ntp.org' na '192.168.0.1'
# rc-update add ntp-client default
# /etc/init.d/ntp-client start
```

Serwer Rsync

Osoby posiadające wiele komputerów z Gentoo w obrębie jednego LAN-u z pewnością wolą pobrać jeden obraz drzewa Portage z Internetu i później udostępnić go dla innych komputerów niż uruchamiać polecenie emerge sync osobno na każdym z nich. Pozwala to zmniejszyć zarówno obciążenie serwerów rsync Gentoo jak i lokalnych komputerów. Konfiguracja jest stosunkowo prosta.

Uwaga: Więcej szczegółów znajduje się w tekście zatytułowanym **Zasady pracy serwerów rsync**.

W związku z tym, że każdy komputer z Gentoo musi mieć zainstalowany program rsync, nie ma potrzeby go teraz instalować. Wystarczy wedytować domyślny plik konfiguracyjny /etc/rsyncd.conf odkomentowując w nim część [gentoo-portage] i dodać opcję address. Pozostałe domyślne ustawienia nie wymagają poprawek.

Listing 6.5: Konfiguracja serwera Rsync

```
pid file = /var/run/rsyncd.pid
use chroot = yes
read only = yes
address = 192.168.0.1

[gentoo-portage]
    path = /mnt/space/portage
    comment = Gentoo Linux Portage tree
    exclude = /distfiles /packages
```

Następnie należy uruchomić odpowiednią usługę (przypominamy, ustawienia domyślne są w porządku).

Listing 6.6: Uruchamianie serwera rsync

```
# /etc/init.d/rsyncd start
# rc-update add rsyncd default
```

Na koniec konfigurujemy pozostałe komputery tak, aby korzystały z naszego routera jako domyślnego serwera Rsync.

Listing 6.7: Ustawienie serwera Rsync w pliku make.conf na pozostałych komputerach

```
SYNC="rsync://192.168.0.1/gentoo-portage"
```

Serwer pocztowy

Czasami miło jest uruchomić na routerze własny prosty serwer poczty (SMTP). Ja go uruchomiłem, aby użytkownicy widzieli swoje maile jako wysłane natychmiastowo i aby cała praca nad ponownym przesyłaniem czy routowaniem została przeniesiona na serwer pocztowy. Warto zwrócić uwagę, że niektórzy usługodawcy Internetowi nie pozwalają na przekazywanie poczty dla kont, które nie są częścią ich sieci (np. Verizon). Wartościową opcją jest możliwość kontrolowania wielkości przesyłanych załączników tak, aby zbyt duże pliki nie blokowały łącza.

Listing 6.8: Konfiguracja SMTP

```
# emerge qmail
Wynik polecenia `hostname` musi być poprawny
# ebuild /var/db/pkg/*-*/qmail-1.03-r*/*.ebuild config
# iptables -I INPUT -p tcp --dport smtp -i ! ${LAN} -j REJECT
# ln -s /var/qmail/supervise/qmail-send /service/qmail-send
# ln -s /var/qmail/supervise/qmail-smtpd /service/qmail-smtpd

# cd /etc
# nano tcp.smtp
Dodaj taki wpis do sekcji allow:
192.168.0.:allow,RELAYCLIENT=""

# tcprules tcp.smtp.cdb rules.tmp < tcp.smtp
# rc-update add svscan default
# /etc/init.d/svscan start
```

Osobiście jestem wielkim fanem programu qmail, ale jego wybór nie jest tu obowiązkowy i można swobodnie wybrać inny MTA. Podczas konfiguracji poczty należy ustawić jako serwer adres 192.168.1.0 i wszystko powinno działać prawidłowo. Więcej informacji na ten temat znajduje się na stronie domowej projektu [Qmail](#).

Kompletny serwer DHCP

Wcześniej skorzystaliśmy z pakietu dnsmasq jako źródła usługi DHCP w naszej sieci. Dla ludzi, którzy opiekują się małymi LAN-ami jest to wystarczające rozwiązanie. Niektórzy jednak mogą być bardziej wymagający i zapragną bardziej wyszukanych możliwości. Takim osobom polecam serwer DHCP z krwi i kości, stworzony przez [ISC](#).

Listing 6.9: Konfiguracja dhcpcd

```
# emerge dhcpcd
# nano /etc/dhcp/dhcpcd.conf
(Przykładowy plik konfiguracyjny)
authoritative;
ddns-update-style interim;
subnet 192.168.0.0 netmask 255.255.255.0 {
    range 192.168.0.100 192.168.0.250;
    default-lease-time 259200;
    max-lease-time 518400;
    option subnet-mask 255.255.255.0;
    option broadcast-address 192.168.0.255;
    option routers 192.168.0.1;
```

```

        option domain-name-servers 192.168.0.1;
}
# nano /etc/conf.d/dhcpd
(Ustawiamy IFACE="eth0")
# rc-update add dhcpcd default
# /etc/init.d/dhcpcd start

```

Jest to minimalna konfiguracja spełniająca te same funkcje co przedstawiony wyżej dnsmasq. W związku z tym, że zmieniliśmy pakiet, który będzie spełniał funkcje serwera DHCP, musimy wyłączyć taką możliwość dla dnsmasq. W tym celu wystarczy zakomentować wpis dhcp-range w pliku /etc/dnsmasq.conf i ponownie uruchomić usługę.

Podłączanie drugiego LAN-u (lub nawet kilku)

Czasem pojawia się potrzeba podłączenia do routera drugiej sieci LAN. Przyczyny tego mogą być różne, a sama procedura jest dość prosta. W poniższym przykładzie zakładamy, że LAN zostanie podłączony za pomocą trzeciej karty sieciowej, eth2.

Przede wszystkim należy skonfigurować interfejs. Informacje na ten temat znajdują się w [listingu 4.1](#), wystarczy zastąpić tam eth0 przez eth2, a zamiast 192.168.0 wpisać 192.168.1.

Następnie należy dokonfigurować dnsmasq, dodając do niego informacje o nowym interfejsie. W tym celu edytujemy plik /etc/conf.d/dnsmasq po raz kolejny i dodajemy linię -i eth2 do zmiennej DNSMASQ_OPTS - możliwe jest użycie kilku parametrów -i. Później edytujemy /etc/dnsmasq.conf i dodajemy kolejną linię z zakresem adresów IP, podobną do tej z [listingu 5.1](#). Posiadanie kilku linii tego typu nie stanowi żadnego problemu.

Na koniec przeglądamy listing 5.2 wdziale o konfiguracji routera i duplikujemy wpisy z -i \${LAN}. Można również zastosować do tego celu drugą zmienną, np. LAN2, żeby trochę uprościć konfigurację.

7. Problemy

Przydatne narzędzia

W przypadku problemów z komunikacją pomiędzy komputerami warto skorzystać z następujących, znajdujących się w kategorii net-analyzer drzewa Portage, narzędzi:

Program Opis

Program	Opis
ethereal	Narzędzie z interfejsem graficznym pozwalające na podglądarkanie danych przesyłanych w sieci w oparciu o zestaw filtrów.
tcpdump	Konsolowy program do podglądarkania ruchu pakietów, również oparty o zestaw filtrów
iptraf	Monitor sieci LAN oparty o biblioteki graficzne ncurses
ettercap	Program do kontroli i monitorowania sieci, również oparty na ncurses

DHCP się nie uruchamia

Pierwsze uruchomienie skryptu startowego init.d dla dhcpcd może się nie udało, w dodatku nie informując co poszło nie tak.

Listing 7.1: Przykład błędu DHCP

```
# /etc/init.d/dhcpcd start
* Setting ownership on dhcp.leases ... [ ok ]
* Starting dhcpcd ... [ !! ]
```

Aby rozwiązać ten problem, wystarczy wiedzieć gdzie dhcpcd wysyła komunikaty o błędach. Wystarczy przejść do /var/log i poczytać logi. W związku z tym, że nazwa pliku z logiem jest różna w zależności od programu logującego działającego na komputerze, radzimy uruchomić grep -R dhcpcd /var/log, aby odnaleźć właściwą lokację. Błąd ten zwykle jest spowodowany literówką w pliku konfiguracyjnym. Można uruchomić dhcpcd -d -f(debug, praca w tle) i sprawdzić na czym dokładnie usługa się psuje.

Niewłaściwy rozmiar MTU

W przypadku dziwnych błędów (np. gdy tylko niektóre strony się wczytują) warto sprawdzić czy nie wystąpił problem z wielkością MTU. Szybki sposób na sprawdzenie tego to wykonanie następującego polecenia:

Listing 7.2: Rozwiązywanie problemów z MTU

```
# iptables -A FORWARD -p tcp --tcp-flags SYN,RST SYN -j TCPMSS
--clamp-mss-to-pmtu
```

Będzie to miało wpływ tylko na nowe połączenia, dlatego należy odświeżyć strony, które wcześniej się nie ladowały w celu przetestowania nowych ustawień. W naprawie może pomóc informacja, że standardowa wartość MTU dla połączenia do 100 Mbit to 1492 Więcej informacji znajduje się w 15 rozdziale tekstu Linux Advanced Routing & Traffic Control HOWTO.

Nie daje się połączyć dwóch komputerów bezpośrednio

Jeśli komputery mają być połączone bezpośrednio, czyli bez udziału huba lub switcha, zwykły kabel sieciowy nie zadziała jeśli nie ma się specjalnego urządzenia (Auto MDI/MDX), które samo wykrywa rodzaj kabla. Konieczne wtedy będzie skorzystanie ze specjalnego rodzaju skrętki nazywanej skrosowaną. Więcej informacji na jej temat znajduje się w Wikipedii.

21. Optymalizacja komplikacji

1. Wstęp

Czym są CFLAGS i CXXFLAGS?

CFLAGS i CXXFLAGS są zmiennymi środowiskowymi, które są używane przy komplikacji kodu przez kompilator gcc. Mówią one jakie przełączniki mają być użyte podczas komplikacji. Zmienna CFLAGS jest dla kodu napisanego w języku C, zaś CXXFLAGS dla kodu napisanego w C++.

Mogą one przysłużyć się do zmniejszenia ilości informacji diagnostycznych dla programu, wzrostu poziomu ostrzeżeń o błędach i optymalizacji wygenerowanego kodu. [Podręcznik GNU gcc](#) zawiera kompletną listę z dostępnymi opcjami i opisem ich działania.

Jak ich używać?

CFLAGS i CXXFLAGS mogą być użyte na dwa sposoby. Po pierwsze przez program przy użyciu plików Makefile wygenerowanych przez automake.

Nie należy tego używać podczas instalacji pakietów z drzewa Portage. Od tego są zmienne CFLAGS i CXXFLAGS w /etc/make.conf. W ten sposób wszystkie pakiety zostaną skompilowane z danymi opcjami.

Listing 1.1: CFLAGS w /etc/make.conf

```
CFLAGS="-march=athlon64 -O2 -pipe"
CXXFLAGS="${CFLAGS}"
```

Jak widać, zmienna CXXFLAGS zawiera wszystkie opcje podane przez CFLAGS. Jest to bezpieczny sposób, który nigdy nie powinien zawieść. Nigdy nie powinno się dodawać innych opcji do CXXFLAGS.

Ważne: Portage nie może użyć CFLAGS na podstawie pakietu, który ma kompilować. Nie ma też dostępnej metody, która mogła by to wymusić. Flagi, które są ustawione w /etc/make.conf zostaną użyte dla wszystkich instalowanych pakietów.

Błędne mniemanie

Kiedy mówi się, że CFLAGS i CXXFLAGS mogą być bardzo efektywne, znaczy to, że produkują mniejszą i/lub szybszą wersję binarną. Ale mogą też osłabić funkcje kodu oraz powiększyć jego rozmiary, spowolnić czas jego wykonywania a nawet spowodować niepowodzenie jego komplikacji!

CFLAGS nie jest receptą na sukces. Nie stworzą systemu, który będzie się uruchamiał szybciej lub też skompilowany kod będzie zajmował mniej miejsca. Dodawanie większej ilości flag w celu optymalizacji zaowocuje efektem odwrotnym. Jest to pewna recepta na niepowodzenie.

W interecie można znaleźć informacje o tym, że agresywne flagi mogą wyrządzić więcej szkód w systemie niż pożytku. Należy pamiętać o tym, że powodem istnienia flag jest określony cel działania. To, że poszczególne opcje są dobre dla kawałka kodu, nie oznacza, że są dobre dla wszystkiego, co będzie instalowane w systemie!

Gotowi?

Skoro już jesteśmy świadomi ryzyka, spójrzmy na kilka bezpiecznych flag dla naszego systemu. Pozwoli to być pomocnym zawsze, kiedy zostanie zgłoszony problem na [Bugzilla](#). Deweloperzy zwykle proszą o ponowną komplikację kodu z minimalnymi flagami, aby dowiedzieć się, czy problem faktycznie istnieje. Agresywne flagi mogą zrujnować kod.

2. Optymalizacja

Podstawy

Celem flag jest stworzenie kodu idealnie dopasowanego do konfiguracji sprzętowej; powinien działać prawidłowo, a zarazem być mały i szybki jak tylko możliwe. Czasami te dwie cechy się wzajemnie wykluczają, więc trzeba trochę poeksperymentować z kombinacjami, aby były prawidłowe. Są one dostępne dla każdej architektury procesora. Agresywne flagi zostaną opisane w późniejszym etapie. Nie omówimy każdej opcji z podręcznika gcc, ponieważ są ich setki. Skupimy się na głównych, najczęściej używanych flagach.

Uwaga: Nie będąc pewnym do czego służy dana flaga, należy odwołać się do odpowiedniej sekcji Podręcznika gcc. W razie dalszych wątpliwości można również sprawdzić listę mailingową gcc.

-march

Najważniejszą opcją jest `-march`. Mówiąc o kompilatorowi jaki powinien być kod wynikowy dla używanej architektury (arch) procesora. Różne rodzaje procesorów mają różne zdolności, wspierają różne instrukcje i mają różne rodzaje wykonywania kodu. Flaga `-march` ma za zadanie stworzenie kodu, który będzie specyficzny dla danego procesora, ze wszystkimi tymi zdolnościami, cechami, instrukcjami itp.

Nawet jeśli zmienna `CHOST` jest ustawiona w pliku `/etc/make.conf` i określa używaną architekturę, flaga `-march` powinna być użyta aby programy mogły zostać zoptymalizowane dla konkretnego procesora.

Aby dowiedzieć się, jaki rodzaj procesora jest w komputerze, którego używamy, należy wykonać polecenie:

Listing 2.1: Analiza informacji o procesorze

```
$ cat /proc/cpuinfo
```

Oglądnijmy `-march` w akcji. Przykład jest dla starszego procesora Pentium III

Listing 2.2: /etc/make.conf: Pentium III

```
CFLAGS="-march=pentium3"
CXXFLAGS="${CFLAGS}"
```

Ten przykład pochodzi z 64-bitowego procesora Sparc.

Listing 2.3: /etc/make.conf: Sparc

```
CFLAGS="-march=ultrasparc"
CXXFLAGS="${CFLAGS}"
```

Dostępne są również flagi `-mcpu` oraz `-mtune`. Jednak powinny zostać one użyte tylko wtedy, gdy opcja `-march` nie jest dostępna. Jaka jest między nimi różnica? `-march` dokładniej precyzuje jakie cechy powinny zostać użyte podczas komplikacji kodu. `-mcpu` stwarza ogólny kod, mało zoptymalizowany dla danego procesora. `-mtune` tworzy kod jeszcze bardziej ogólny. Należy użyć `-march` zawsze kiedy jest dostępny. Dla architektur takich jak PowerPC lub Alpha musi zostać użyty `-mcpu`.

Uwaga: Dla konkretniejszych ustawień `-march` należy przeczytać rozdział „5.e” Podręcznika instalacji Gentoo dla odpowiedniej architektury. Polecamy również przeczytać listę podręcznika gcc o opcjach specyficznych dla danej architektury gdzie jest dokładnie wyjaśniona różnica pomiędzy `-march`, `-mcpu` i `-mtune`. Jest to bardzo pomocne przy precyzowaniu jakie opcje `-march` powinny zostać użyte. Zwłaszcza, że na niektórych architekturach (takich jak x86) flaga `-mcpu` jest przestarzała i powinno się użyć `-mtune` zamiast niej.

-O

Nastecną opcją jest `-O`. Kontroluje ona całkowity poziom optymalizacji. Może ona spowodować, że kod będzie się komplikował nieco dłużej i zostanie użyte nieco więcej pamięci wraz z jej podnoszeniem.

Jest pięć opcji `-O`: `-O0`, `-O1`, `-O2`, `-O3` oraz `-Os`. W `/etc/make.conf` powinna być użyta tylko jedna.

Z wyjątkiem `-O0`, każda następna opcja `-O` aktywuje kilka dodatkowych flag. O tym jakie flagi są aktywowane przy każdym poziomie `-O` i co powodują, można się dowiedzieć po przeczytaniu rozdziału o opcjach optymalizacji strony manuala gcc.

Przeanalizujmy każdy poziom optymalizacji:

- `-O0`: Ten poziom (jest to litera "O" za którą jest cyfra zero) kompletnie wyłącza optymalizację i jest domyślna kiedy opcja `-O` nie jest określona w zmiennych `CFLAGS` i `CXXFLAGS`, co nie jest generalnie pożądane.
- `-O1`: Bardzo podstawowy poziom optymalizacji. Kompilator będzie próbował budować szybszy i mniejszy kod bez tracenia na czasu. Jest to naprawdę podstawowy poziom i nie powinien stanowić problemów podczas komplikacji.
- `-O2`: Poziom wyżej od `-O1`. Jest to zalecany poziom optymalizacji jeżeli nie mamy specjalnych potrzeb (takich jak `-Os`, co zostanie wyjaśnione wkrótce). `-O2` do `-O1` doda kilka nowych flag. Z opcją `-O2` komplikator będzie próbował zwiększyć wydajność kodu nie zwracając uwagi na rozmiar kodu wynikowego przy długim czasie komplikacji.
- `-O3`: Najwyższy możliwy poziom optymalizacji i zarazem najbardziej ryzykowny. Z tą opcją komplikator będzie tracił dużo czasu na stworzenie kodu wynikowego. Generalnie opcja nie powinna być używana na systemach używających gcc w wersji 4.x. Zachowanie gcc znacznie się zmieniło od wersji 3.x, gdzie opcja `-O3` miała prowadzić do trochę szybszego wykonania niż przy `-O2`. Przy wersji kompilatora gcc 4.x, komplikowany z użyciem `-O3` kod będzie miał wielki rozmiar, a jego wykonanie będzie wymagało użycia większej ilości pamięci. Będzie również znacznie zwiększał przypadki niepowodzenia komplikacji i nieoczekiwane zachowanie programów. **Używanie `-O3` nie jest zalecane dla gcc 4.x.**
- `-Os`: Ten poziom będzie optymalizował kod pod względem wielkości. Aktywuje on wszystkie flagi z poziomu `-O2`, które nie biorą udziału w zwiększeniu wygenerowanego kodu. Jest to przydatne dla maszyn, które mają bardzo ograniczoną przestrzeń dyskową lub z procesorami o bardzo małej wielkości cache.

Jak zostało wcześniej wspomniane, -O2 jest rekomendowanym poziomem optymalizacji. Jeśli pakiet zawiedzie podczas komplikacji, należy się upewnić, że nie jest używana opcja -O3. Należy wtedy spróbować z niższym poziomem optymalizacji, takim jak -O1 lub -Os.

-pipe

-pipe jest bezpieczną flagą. Nie ma ona wpływu na generowany kod, lecz przynosi zysk w prędkości komplikacji. Każe kompilatorowi używa potoków zamiast plików tymczasowych w różnych etapach komplikacji.

-fomit-frame-pointer

Jest to bardzo powszechna flaga stworzona do zmniejszania wielkości generowanego kodu. Jest ona włączona w każdym poziomie -O (poza -O0) na architekturach gdzie nie przeszkadza to przy debugowaniu (takich jak np. x86-64). Ponieważ strona podręcznika GNU gcc nie podaje wszystkich architektur, na których -O wymusza działanie tej flagi, czasem trzeba ją dodać własnoręcznie. Używanie tej flagi może powodować problemy podczas debugowania kodu.

Szczególne problemy sprawiają aplikacje napisane w Javie, chociaż nie jest ona jedynym językiem napiętnowanym przez tę flagę. Trzeba mieć na uwadze, że mimo tego, że flaga ogólnie pomaga, może sprawić debugowanie kodu znacznie bardziej trudnym. Jeśli nie zamierzamy poddawać kodu operacjom debugowania i nie mamy w zmiennej CFLAGS innych flag, takich jak -ggdb (ani nie komplujemy z włączoną flagą USE="debug"), warto dodać -fomit-frame-pointer.

Ważne: Nie należy łączyć -fomit-frame-pointer z podobnie działającą flagą -momit-leaf-frame-pointer. Użycie drugiej flagi jest niepotrzebne, ponieważ -fomit-frame-pointer wykonuje swoją pracę prawidłowo. Ponadto, -momit-leaf-frame-pointer okazuje się negatywnie wpływać na jakość kodu.

-msse, -msse2, -msse3, -mmmx, -m3dnow

Te flagi uruchamiają ustawienia instrukcji [SSE](#), [SSE2](#), [SSE3](#), [MMX](#) i [3DNow!](#) dla architektur x86 i x86-64. Jest to użyteczne przede wszystkim przy multimedialach, grach i zadaniach, wymagających intensywnych obliczeń matematycznych. Funkcje te są dostępne w nowszych modelach procesorów.

Ważne: Należy się upewnić, że dany procesor wspiera powyższe flagi. Można się tego dowiedzieć poprzez cat /proc/cpuinfo. Wyjście tego polecenia będzie zawierało wszystkie dodatkowe ustawienia. Trzeba mieć na uwadze, że pni jest inną nazwą dla SSE3.

Generalnie nie ma potrzeby dodawania jakiekolwiek z tych flag, dopóki jest ustawiona odpowiednia opcja -march (np. -march=nocona zawiera -msse3). Kilka warych uwag wyjątków to nowsze procesory VIA i AMD64, które wspierają instrukcje niezawarte w -march (takie jak SSE3). Dla takich procesorów będzie wymagane dodanie flag, które będą odpowiadały wynikowi cat /proc/cpuinfo.

Uwaga: Należy zapoznać się z [listą flag specyficznych tylko dla architektur x86 i x86-64](#), aby wiedzieć co aktywuje każda z tych ustawień. Gdy instrukcja jest podana, wtedy nie ma potrzeby jej wyszczególnienia. Będzie ona domyślnie włączona przy odpowiedniej flagie -march.

3. FAQ Optymalizacji

Ale można osiągnąć lepsze wyniki przy użyciu -funroll-loops -fomg-optimize!

Niestety. Może się tylko wydawać, ponieważ ktoś nas przekonał, że wydajność rośnie z liczbą użytych flag. Agresywne flagi mogą tylko uszkodzić aplikacje. Nawet strona manuala gcc mówi, że użycie -funroll-loops i -funroll-all-loops tworzy kod większy i bardziej powolny. Po -ffast-math, -fforce-mem, -fforce-addr, są to kolejne mylnie używane flagi.

Prawdę jest, że są to bardzo agresywne flagi. Można się o tym przekonać w bardzo łatwy sposób: odwiedzając [Forum Gentoo](#) oraz Bugzilla. Szybko zauważymy, że nie robią one nic dobrego!

Nie należy ich używać, ponieważ stworzą one spuchnięty kod o wielkich rozmiarach i słabej wydajności. Sprawią, że system będzie działał na krawędzi, a nasze bugi zostaną oznaczone jako INVALID lub WONTFIX.

Nie potrzeba tak niebezpiecznych flag. **Nie należy ich używać.** Lepsze od tego jest użycie podstawowych flag: -march, -O i -pipe.

Co z poziomami -O wyższymi niż 3?

Niektórzy użytkownicy chwalą się lepszą wydajnością przy użyciu -O4, -O9 itp. Prawdę jednak jest to, że poziomy wyższe niż 3 nie mają żadnego wpływu. Kompilator akceptuje CFLAGS takie jak -O4, lecz nie wykorzystuje ich przy budowie kodu. Jest to równoznaczne z użyciem -O3, nic więcej.

Więcej dowodów? Przeanalizuj kod źródłowy gcc:

Listing 3.1: Kod źródłowy -O

```
if (optimize >= 3)
{
    flag_inline_functions = 1;
    flag_unswitch_loops = 1;
    flag_gcse_after_reload = 1;
    /* Allow even more virtual operators. */
    set_param_value ("max-aliased-vops", 1000);
    set_param_value ("avg-aliased-vops", 3);
}
```

Jak widać, każda wartość większa niż 3 jest traktowana jak -O3.

Co ze zbędnymi flagami?

Opcje CFLAGS i CXXFLAGS, które są zawarte w różnych poziomach -O są zbędne w /etc/make.conf. Czasami są one ignorowane, a czasami filtrowane lub zastępowane.

Filtracje i zastępowanie flag jest wykorzystywane przez wiele ebuildów w drzewie Portage. Zazwyczaj jest to robione z powodu niepowodzenia przy komplikacji z określonym poziomem optymalizacji -O lub kiedy kod jest zbyt wrażliwy na wszelkie dodatkowe flagi. Ebuild będzie filtrował niektóre CFLAGS i CXXFLAGS lub zastąpi -O innym poziomem.

[Podręcznik Dewelopera Gentoo](#) tworzy szkic gdzie i jak filtrowanie i zastępowanie flag działa.

Można ominąć filtrowania -O poprzez niepotrzebne dodawanie flag pewnych poziomów optymalizacji, takich jak -O3

Listing 3.2: Zbędne opcje w CFLAGS

```
CFLAGS="-O3 -finline-functions -funswitch-loops"
```

Nie jest to mądre. CFLAGS jest filtrowane z konkretnego powodu! Kiedy flagi są filtrowane, znaczy to, że są niebezpieczne do budowy pakietu. Naprawdę, nie jest bezpieczne budowanie całego systemu z -O3 jeśli jakaś flaga z tego poziomu optymalizacji będzie tworzyła problemy z danymi pakietami. Zaufaj deweloperom. Filtrowanie i zastępowanie flag jest robione dla bezpieczeństwa naszych danych. Jeśli ebuild określa alternatywną flagę, nie należy tego obchodzić w żaden sposób!

Po zbudowaniu pakietu z nieakceptowanymi flagami, najczęściej pogłębiamy się w problemach. Kiedy zgłaszymy błąd na Bugzilla, użyte flagi z /etc/make.conf są widoczne. Zostaniemy wtedy powiadomieni, aby przebudować dany pakiet bez niepotrzebnych flag. Można sobie tego oszczędzić poprzez ich nieużywanie i niezakładanie, że wie się lepiej niż deweloperzy.

Co z LDFLAGS?

Nie należy ich używać. Można usłyszeć, że mogą podnieść czas uruchamiania aplikacji lub zmniejszyć ich rozmiar. Jednak LDFLAGS najczęściej powodują zepsucie się danej aplikacji. Nie są one wspierane, więc prawdopodobnie zgłoszony bug zostanie oznaczony jako INVALID. Ostatecznie trzeba będzie ponownie komplikować pakiet bez ustawionych LDFLAGS.