

# Administrowanie systemami sieciowymi

## Sprawozdanie z laboratorium

Data	Tytuł zajęć	Uczestnicy
02.12.2020 11:15	Włączanie i wyłączanie systemu Linux; Podstawowe usługi systemu Linux; Zarządzanie programami i usługami	Bartosz Rodziejwicz (226105)

### Opis środowiska

Zajęcia laboratoryjne z części nt. systemu Linux zostały wykonane na maszynie wirtualnej postawionej z wykorzystaniem VirtualBox. Zainstalowana w maszynie dystrybucja to Manjaro 20.2 ze środowiskiem (DE) KDE Plasma. Do zajęć użyta została czysta instalacja systemu po doinstalowaniu najnowszych aktualizacji pakietów.

### Przebieg laboratorium

Zapoznać się z narzędziami zarządzania pakietami dostępnymi w wybranej dystrybucji. Zainstalować w systemie menedżer plików Midnight Commander ( `mc` ). Jeśli program ten jest już zainstalowany, wybrać inny program do zainstalowania.

#### Zapoznać się z narzędziami zarządzania pakietami dostępnymi w wybranej dystrybucji.

W dystrybucji Manjaro dostępne są dwa narzędzia do zarządzania pakietami - `pacman` i `pamac` .

Pierwszy z nich jest starszym narzędziem wywodzącym się z dystrybucji Arch (z której Manjaro samo się wywodzi). `pacman` jest narzędziem CLI, czyli programem który można używać tylko z terminala (istnieją do niego różne GUI, jak np. `octopi` , jednak sam `pacman` takiego nie udostępnia). `pacman` w Manjaro umożliwia instalacje pakietów z oficjalnych repozytoriów systemu.

`pamac` to nowe narzędzie do zarządzania pakietów stworzone na potrzeby Manjaro. Posiada bardziej przyjazny interfejs CLI ("łatwiejsze" komendy) oraz samemu udostępnia GUI. GUI `pamac` poza podstawowymi operacjami, jak instalowanie i usuwanie pakietów posiada również możliwość przeglądania pakietów z repo wraz z możliwością czytania opisów i przeglądania zrzutów ekranu. Pozwala to uznać ją za aplikację typu sklep, gdzie poza samą instalacją można odkrywać nowe oprogramowanie w przyjemnym, nowoczesnym interfejsie. Brak takiego miejsca był częstym zrzutem dlaczego ta dystrybucja nie nadaje się dla początkujących użytkowników Linuxa. `pamac` poza instalacją oprogramowania z oficjalnych repo systemu umożliwia również instalację z AURa (wcześniej do obsługi AUR na Manjaro konieczne było doinstalowanie pakietu lub ręczna obsługa), a także trwają prace nad obsługą Snapa i Flatpacka.

W dystrybucji Manjaro istnieją dwa główne źródła pakietów:

- oficjalne repozytoria
- AUR - Arch User Repository

Oficjalne repo systemowe to repozytoria z najpopularniejszymi i systemowymi gotowymi pakietami. Repozytoria te bazują na repozytoriach dystrybucji Arch, jednak posiadają bardziej przetestowane pod kątem stabilności i niezawodności oprogramowanie Arch i Manjaro są dystrybucjami typu rolling release, czyli zawsze posiadają najnowszą wersję danego pakietu w repo. Manjaro w swoich repo dłużej testuje oprogramowanie, więc aktualizacje ukazują się z lekkim opóźnieniem do Arch, zapewniając lepszą stabilność.

AUR to zbiór pakietów zarządzany przez społeczność Arch (jak i pochodnych). Sam AUR przechowuje tylko skrypty do przygotowania i instalacji pakietu. Same źródła (lub binarka) programu najczęściej pobierana jest z oficjalnego źródła wydawcy. W AURze można znaleźć praktycznie wszystkie pakiety jakie wyszły na Linuxa czyniąc instalacje oprogramowania z poza oficjalnych repo bardzo łatwą. Nie jest to jednak rozwiązanie idealne. Przy korzystaniu z AURa trzeba zachować czujność i zrozumieć skrypty instalacyjne, ponieważ zdarzały się przypadki niebezpiecznych komend w tych skryptach. Skrypty te wykorzystywane są przez `makepkg` , który służy do przygotowywania pakietów do Arch. Użycie AURa możliwe jest ręcznie (samodzielne pobranie skryptów i potrzebnych binarek/źródeł oraz wykonanie `makepkg` ) lub za pomocą różnych helperów AUR, jak wspomniany przeze mnie, od niedawna natywny dla Manjaro, `pamac` .

Zainstalować w systemie menedżer plików Midnight Commander ( `mc` ). Jeśli program ten jest już zainstalowany, wybrać inny program do zainstalowania.

Zainstalowanie pakietu `mc` możliwe jest za pomocą komendy `pamac install mc`. Później `pamac` pyta się o instalację opcjonalnych zależności, hasło użytkownika (user musi być na liście sudoers) oraz prosi o potwierdzenie "transakcji".

```
[baatochan@baatochan-virtualbox ~]$ pamac install mc
Preparing...
Choose optional dependencies for mc:
 1:  aspell:  spelling corrections
 2:  cabextract:  ucab extfs
 3:  cvs:  CVS support
 4:  mtools:  a+ extfs
 5:  python-boto:  s3+ extfs
 6:  python-pytz:  s3+ extfs
 7:  unace:  uace extfs
 8:  unarj:  uarj extfs
 9:  unrar:  urar extfs
10:  unzip:  open zip archives
11:  zip:  uzip extfs

Enter a selection (default=none): 1,2,3,4,5,6,7,8,9,10,11

==== AUTHENTICATING FOR org.manjaro.pamac.commit ====
Authentication is required to install, update, or remove packages
Authenticating as: Baatochan (baatochan)
Password:
==== AUTHENTICATION COMPLETE ====
Synchronizing package databases...
Resolving dependencies...
Checking inter-conflicts...
To install (12):
unzip          6.0-14          extra          142.8 kB
python-boto    2.49.0.20190327-3 community      1.4 MB
mc             4.8.25-3        community      2.0 MB
unarj          2.63a-6         community      15.0 kB
zip            3.0-9           extra          174.9 kB
cvs            1.11.23-13      extra          566.5 kB
unrar          1:6.0.2-1       extra          144.3 kB
unace          2.5-11          community      58.5 kB
aspell         0.60.8-2        extra          697.7 kB
mtools         4.0.26-1        extra          214.6 kB
python-pytz    2020.4-1        community      41.5 kB
cabextract     1.9.1-2         community      36.5 kB

Total download size: 5.5 MB
Total installed size: 26.9 MB
Apply transaction ? [y/N] y
```

Na zrzucie powyżej widzimy instalację pakietu `mc` z wszystkimi opcjonalnymi zależnościami.

Korzystając z `dpkg` lub `APT` sprawdzić, czy w systemie jest zainstalowany program `httpd` (serwer WWW Apache) oraz określić jakie pliki wchodzi w jego skład. Określić, na jakich poziomach pracy ten program (usługa) jest uruchomiony.

Aby sprawdzić, czy dana paczka jest zainstalowana można użyć komendy `pamac search <nazwa_paczki>`. Poniżej widzimy, że pakiet `apache` jest dostępny w systemie. *Note: W dystrybucji Manjaro nazwa pakietu serwera http apache to `apache`, a nie `httpd`.*

```
[baatochan@baatochan-virtualbox ~]$ pamac search apache
mod_passenger 6.0.7-1 community
  Passenger apache module
ant            1.10.9-1 extra
  Java based build tool
php-apache     7.4.13-1 extra
  Apache SAPI for PHP
dbeaver-plugin-apache-poi 3.16.0-4 community
  DBeaver library for Microsoft Office documents
certbot-apache 1.9.0-1 community
  Apache plugin for Let's Encrypt client
apache         [Installed] 2.4.46-3 extra
  A high performance Unix-based HTTP server
[baatochan@baatochan-virtualbox ~]$
```

Listę plików danej paczki można sprawdzić za pomocą komendy `pamac list --files <nazwa_paczki>`. Lista plików serwera Apache jest bardzo długa, więc pokazuje tylko jej początek. Listę plików można sprawdzić również dla niezainstalowanej paczki.

```
[baatochan@baatochan-virtualbox ~]$ pamac list --files apache
/etc/httpd/conf/extra/httpd-autoindex.conf
/etc/httpd/conf/extra/httpd-dav.conf
/etc/httpd/conf/extra/httpd-default.conf
/etc/httpd/conf/extra/httpd-info.conf
/etc/httpd/conf/extra/httpd-languages.conf
/etc/httpd/conf/extra/httpd-manual.conf
/etc/httpd/conf/extra/httpd-mpm.conf
/etc/httpd/conf/extra/httpd-multilang-errordoc.conf
/etc/httpd/conf/extra/httpd-ssl.conf
/etc/httpd/conf/extra/httpd-userdir.conf
/etc/httpd/conf/extra/httpd-vhosts.conf
/etc/httpd/conf/extra/proxy-html.conf
/etc/httpd/conf/httpd.conf
/etc/httpd/conf/magic
/etc/httpd/conf/mime.types
/etc/httpd/modules
/etc/logrotate.d/httpd
/usr/bin/ab
/usr/bin/apachectl
/usr/bin/apxs
/usr/bin/checkgid
/usr/bin/dbmmanage
/usr/bin/envvars
/usr/bin/envvars-std
/usr/bin/fcgistarter
/usr/bin/htcacheclean
/usr/bin/htdbm
/usr/bin/htdigest
/usr/bin/htpasswd
/usr/bin/httpd
/usr/bin/httxt2dbm
/usr/bin/logresolve
/usr/bin/rotatelog
/usr/bin/suexec
/usr/include/httpd/ap_compat.h
/usr/include/httpd/ap_config.h
/usr/include/httpd/ap_config_auto.h
/usr/include/httpd/ap_config_layout.h
/usr/include/httpd/ap_expr.h
/usr/include/httpd/ap_hooks.h
/usr/include/httpd/ap_listen.h
/usr/include/httpd/ap_mmn.h
/usr/include/httpd/ap_mpm.h
/usr/include/httpd/ap_provider.h
/usr/include/httpd/ap_regex.h
/usr/include/httpd/ap_regkey.h
/usr/include/httpd/ap_release.h
/usr/include/httpd/ap_slotmem.h
/usr/include/httpd/ap_socache.h
```

Pakiet apache do działania wykorzystuje własną control grupę, na poziomie systemowym (roota).

```
[baatochan@baatochan-virtualbox ~]$ systemctl status httpd
● httpd.service - Apache Web Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
   Active: inactive (dead)

[baatochan@baatochan-virtualbox ~]$ systemctl start httpd
[baatochan@baatochan-virtualbox ~]$ systemctl status httpd
● httpd.service - Apache Web Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
   Active: active (running) since Sat 2020-12-12 14:04:47 CET; 1s ago
     Main PID: 82280 (httpd)
        Tasks: 82 (limit: 4698)
       Memory: 6.1M
       CGroup: /system.slice/httpd.service
              └─82280 /usr/bin/httpd -k start -DFOREGROUND
                 └─82281 /usr/bin/httpd -k start -DFOREGROUND
                    └─82282 /usr/bin/httpd -k start -DFOREGROUND
                       └─82283 /usr/bin/httpd -k start -DFOREGROUND

Dec 12 14:04:47 baatochan-virtualbox systemd[1]: Started Apache Web Server.
Dec 12 14:04:47 baatochan-virtualbox httpd[82280]: AH00558: httpd: Could not reliably determine t
[baatochan@baatochan-virtualbox ~]$
```

Zapoznać się z konfiguracją menu startowego systemu Linux ( `grub` ). Zmienić domyślnie ładowany system na uruchamianie testu pamięci. Po przetestowaniu zmian przywrócić poprzedni stan.

Konfiguracja GRUBa wykonywana jest z poziomu pliku `/etc/default/grub`. Opis odpowiednich opcji dostępny jest w [dokumentacji GRUBa](#). Do zmiany domyślnie bootowanego wpisu w grub konieczna będzie również znajomość kolejności wpisów. Możliwe jest to poprzez np. analizę pliku

/boot/grub/grub.cfg. W moim przypadku system operacyjny jest na pozycji 0, natomiast `memtest` na pozycji 2.

```
GRUB_DEFAULT=saved
GRUB_TIMEOUT=5
GRUB_TIMEOUT_STYLE=hidden
GRUB_DISTRIBUTOR="Manjaro"
GRUB_CMDLINE_LINUX_DEFAULT="quiet apparmor=1 security=apparmor udev.log_priority=3"
GRUB_CMDLINE_LINUX=""

# If you want to enable the save default function, uncomment the following
# line, and set GRUB_DEFAULT to saved.
GRUB_SAVEDEFAULT=true

# Preload both GPT and MBR modules so that they are not missed
GRUB_PRELOAD_MODULES="part_gpt part_msdos"

# Uncomment to enable booting from LUKS encrypted devices
#GRUB_ENABLE_CRYPTODISK=y

# Uncomment to use basic console
GRUB_TERMINAL_INPUT=console

# Uncomment to disable graphical terminal
#GRUB_TERMINAL_OUTPUT=console

# The resolution used on graphical terminal
# note that you can use only modes which your graphic card supports via VBE
# you can see them in real GRUB with the command 'videoinfo'
GRUB_GFXMODE=auto

# Uncomment to allow the kernel use the same resolution used by grub
GRUB_GFXPAYLOAD_LINUX=keep

# Uncomment if you want GRUB to pass to the Linux kernel the old parameter
# format "root=/dev/xxx" instead of "root=/dev/disk/by-uuid/xxx"
#GRUB_DISABLE_LINUX_UUID=true

# Uncomment to disable generation of recovery mode menu entries
GRUB_DISABLE_RECOVERY=true

# Uncomment and set to the desired menu colors. Used by normal and wallpaper
# modes only. Entries specified as foreground/background.
GRUB_COLOR_NORMAL="light-gray/black"
GRUB_COLOR_HIGHLIGHT="green/black"

# Uncomment one of them for the gfx desired, a image background or a gfxtheme
#GRUB_BACKGROUND="/usr/share/grub/background.png"
GRUB_THEME="/usr/share/grub/themes/manjaro/theme.txt"

# Uncomment to get a beep at GRUB start
#GRUB_INIT_TUNE="480 440 1"
```

Wyżej widać domyślny plik konfiguracyjny GRUB w tej dystrybucji. Domyślnym ustawieniem jest ukryty GRUB przy starcie oraz zapamiętywanie ostatnio bootowanej pozycji (domyślnie bootowana pozycja to ostatnio wybrana).

Aby zmienić na domyślne bootowanie `memtest` konieczne będzie usunięcie (zakomentowanie znakiem `#`) opcji `GRUB_SAVEDEFAULT`, a dopisanie opcji `GRUB_DEFAULT="2"` (2 to pozycja `memtest`).

Dodatkowo zmieniona zostaje pozycja `GRUB_TIMEOUT_STYLE=hidden`, na `countdown`, aby GRUB zawsze był widoczny podczas bootowania. Ułatwi to powrót do pozycji 0, przy używaniu systemu w maszynie wirtualnej.

```
GRUB_DEFAULT=saved
GRUB_TIMEOUT=5
GRUB_TIMEOUT_STYLE=countdown
GRUB_DISTRIBUTOR="Manjaro"
GRUB_CMDLINE_LINUX_DEFAULT="quiet apparmor=1 security=apparmor udev.log_priority=3"
GRUB_CMDLINE_LINUX=""

# If you want to enable the save default function, uncomment the following
# line, and set GRUB_DEFAULT to saved.
#GRUB_SAVEDEFAULT=true
GRUB_DEFAULT="2"
```

Aby zmiana konfiguracji została zapisana, poza zmianą ustawień w tych plikach konieczne jest jeszcze uruchomienie komendy `grub-mkconfig -o /boot/grub/grub.cfg` z uprawnieniami roota.

```
[baatochan@baatochan-virtualbox ~]$ sudo grub-mkconfig -o /boot/grub/grub.cfg
[sudo] password for baatochan:
Generating grub configuration file ...
Found theme: /usr/share/grub/themes/manjaro/theme.txt
Found linux image: /boot/vmlinuz-5.9-x86_64
Found initrd image: /boot/intel-ucode.img /boot/initramfs-5.9-x86_64.img
Found initrd fallback image: /boot/initramfs-5.9-x86_64-fallback.img
Found memtest86+ image: /boot/memtest86+/memtest.bin
done
[baatochan@baatochan-virtualbox ~]$
```



Po reboocie komputera i przejściu licznika automatycznie uruchamia się `memtest`.

```
Memtest86 5.01 | Intel(R) Core(TM) i5-4690K CPU @ 3.50GHz
CLK: 3491 MHz (32b Mode) | Pass %
L1 Cache: 32K 232749 MB/s | Test %
L2 Cache: 256K 52897 MB/s | Test #
L3 Cache: 6144K 64652 MB/s | Testing:
Memory : 4096M | Pattern: | Time: 0:00:00
-----
Core#: | Chipset : Unknown
State: | Memory Type : Unknown
Cores: Active / Total (Run: All) | Pass: 0 Errors: 0
-----
Press F1 to enter Fail-Safe Mode
==> Press F2 to force Multi-Threading (SMP) <==
(ESC)exit (c)configuration (SP)scroll_lock (CR)scroll_unlock
```

Po wyłączeniu `memtest` a i wybootowaniu systemu, aby przywrócić konfigurację należy cofnąć zmiany w pliku `/etc/default/grub` oraz ponownie uruchomić `grub-mkconfig`.

Wykorzystując polecenie `at` wywołać zadanie, którego czas uruchomienia zostanie odłożony o 10 minut od chwili wydania polecenia `at`.

Dystrybucja Manjaro nie posiada domyślnie zainstalowanego pakietu `at`, więc konieczne było jego doinstalowanie.

Zaplanowanie zadania na 10 minut w przyszłość wykonuje się komendą `at now +10 minutes`, po czym podaje komendy, które mają się wykonać. Zaplanowane komendy to:

```
echo "Bartosz Rodziewicz, 226105" >> info.txt
date >> info.txt
w >> info.txt
```

Zakończenie planowania następuje po podaniu EOF, czyli po wciśnięciu `Ctrl+D`.

```
[baatochan@baatochan-virtualbox ~]$ at now +10 minutes
warning: commands will be executed using /bin/sh
at Sat Dec 12 15:26:00 2020
at> echo "Bartosz Rodziewicz, 226105" >> info.txt
at> date >> info.txt
at> w >> info.txt
at> <EOT>
job 2 at Sat Dec 12 15:26:00 2020
[baatochan@baatochan-virtualbox ~]$
```

Poniżej widać że zadanie się wykonało (dwa pierwsze wpisy były pisane z ręki by sprawdzić czy komendy działają).

```
Bartosz Rodziewicz, 226105
Sat 12 Dec 2020 15:11:23 CET
15:11:23 up 13 min, 5 users, load average: 0.16, 0.18, 0.18
USER TTY LOGIN@ IDLE JCPU PCPU WHAT
baatocha tty1 14:58 1:37m 16.61s 0.08s /usr/bin/startplasma-x11
baatocha pts/0 14:58 13:15 0.00s 0.74s /usr/bin/kded5
baatocha pts/1 14:58 13:10 0.02s 0.02s /bin/bash
baatocha pts/2 14:58 3:00s 0.04s 0.00s w
baatocha pts/3 15:09 1:05 0.02s 0.02s /bin/bash
Bartosz Rodziewicz, 226105
Sat 12 Dec 2020 15:11:35 CET
15:11:35 up 13 min, 5 users, load average: 0.13, 0.18, 0.18
USER TTY LOGIN@ IDLE JCPU PCPU WHAT
baatocha tty1 14:58 1:37m 16.74s 0.08s /usr/bin/startplasma-x11
baatocha pts/0 14:58 13:27 0.00s 0.74s /usr/bin/kded5
baatocha pts/1 14:58 13:22 0.02s 0.02s /bin/bash
baatocha pts/2 14:58 7:00s 0.05s 0.00s w
baatocha pts/3 15:09 1:17 0.02s 0.02s /bin/bash
Bartosz Rodziewicz, 226105
Sat 12 Dec 2020 15:26:00 CET
15:26:00 up 28 min, 5 users, load average: 0.14, 0.17, 0.16
USER TTY LOGIN@ IDLE JCPU PCPU WHAT
baatocha tty1 14:58 1:51m 31.05s 0.08s /usr/bin/startplasma-x11
baatocha pts/0 14:58 27:52 0.00s 0.95s /usr/bin/kded5
baatocha pts/1 14:58 27:47 0.02s 0.02s /bin/bash
baatocha pts/2 14:58 16:00s 0.09s 0.09s /bin/bash
baatocha pts/3 15:09 9:52 0.02s 0.00s /bin/bash
info.txt (END)
```

Wykorzystując polecenie `atq` sprawdzić jakie zadania oczekują na wykonanie. Wcześniej zaplanować kilka zadań (dowolnych).

```
[baatochan@baatochan-virtualbox ~]$ atq
3          Sat Dec 12 15:26:00 2020 a baatochan
2          Sat Dec 12 15:26:00 2020 a baatochan
6          Sun Dec 13 15:22:00 2020 a baatochan
5          Sat Dec 12 16:22:00 2020 a baatochan
4          Sat Dec 12 15:42:00 2020 a baatochan
```

Wykorzystując polecenie `atrm`, usunąć jedno z utworzonych wcześniej zadań.

```
[baatochan@baatochan-virtualbox ~]$ atrm 6
[baatochan@baatochan-virtualbox ~]$ atq
3          Sat Dec 12 15:26:00 2020 a baatochan
2          Sat Dec 12 15:26:00 2020 a baatochan
5          Sat Dec 12 16:22:00 2020 a baatochan
4          Sat Dec 12 15:42:00 2020 a baatochan
[baatochan@baatochan-virtualbox ~]$
```

Sprawdzić jakie zadania są aktualnie przeznaczone do wykonywania z wykorzystaniem `crontab`.

```
[baatochan@baatochan-virtualbox ~]$ crontab -l
no crontab for baatochan
[baatochan@baatochan-virtualbox ~]$ sudo crontab -l
[sudo] password for baatochan:
no crontab for root
[baatochan@baatochan-virtualbox ~]$
```

Utworzyć nowy plik z zadaniami dla polecenia `crontab`, o nazwie "`nowy_plik_crontab`". W pliku tym ma się znaleźć linia z zadaniem, które będzie się uruchamiać 15 minut po każdej pełnej godzinie i zapisywać listę uruchomionych procesów do pliku:

`procesy_<godzina>_<data>.log` w katalogu domowym. Następnie należy wywołać polecenie `crontab` wykorzystując utworzony plik (`nowy_plik_crontab`).

Do wykonania zadania plik `new_file_crontab` ma następującą zawartość:

```
15 * * * * /usr/bin/ps -aux > /home/baatochan/processes_`date +%H-%M-%S`_`date +%Y-%m-%d`.log 2>&1
```

gdzie:

- `15 * * * *` - informacja, że polecenie ma zostać wykonane 15 min po każdej pełnej godzinie,
- `/usr/bin/ps -aux` - komenda listująca procesy,
- `> /home/baatochan/processes_`date +%H-%M-%S`_`date +%Y-%m-%d`.log` - przekierowanie wyjścia do pliku `proceses_data_godzina.log` zapisanego w katalogu domowym; komendy `date` muszą być zawarte w ```, ponieważ są to komendy w komendzie, a dodatkowo dla zadania cron wszystkie `%` muszą być wyescapowane,
- `2>&1` - przekierowanie również wyjścia błędów do pliku (tam gdzie standardowe wyjście).

Uruchomienie zadania polega na wykonaniu komendy `crontab new_file_crontab`.

```
[baatochan@baatochan-virtualbox ~]$ cat new_file_crontab
15 * * * * /usr/bin/ps -aux > /home/baatochan/processes_`date +%H-%M-%S`_`date +%Y-%m-%d`.log 2>&1
[baatochan@baatochan-virtualbox ~]$ crontab new_file_crontab
[baatochan@baatochan-virtualbox ~]$ crontab -l
15 * * * * /usr/bin/ps -aux > /home/baatochan/processes_`date +%H-%M-%S`_`date +%Y-%m-%d`.log 2>&1
[baatochan@baatochan-virtualbox ~]$ ls
Desktop          processes_16-31-58_2020-12-12.log  processes_16-56-01_2020-12-12.log
Documents        processes_16-40-17_2020-12-12.log  processes_16-57-01_2020-12-12.log
Downloads        processes_16-42-02_2020-12-12.log  processes_16-58-01_2020-12-12.log
info.txt         processes_16-44-01_2020-12-12.log  processes_16-59-01_2020-12-12.log
Music            processes_16-45-01_2020-12-12.log  Public
new_file_crontab processes_16-46-01_2020-12-12.log  Templates
Pictures         processes_16-52-01_2020-12-12.log  Videos
processes_16-25-06_2020-12-12.log  processes_16-53-02_2020-12-12.log
processes_16-25-52_2020-12-12.log  processes_16-55-01_2020-12-12.log
```

Na zrzucie powyżej widać zawartość pliku, aktywację zadania oraz efekt zadania. Działanie komendy zostało sprawdzone na zadaniu cron uruchamianym co minutę (`* * * * *`) oraz na wykonaniach ręcznych.

Usunąć wszystkie zadania, które aktualnie są przeznaczone do wykonania.

```
[baatochan@baatochan-virtualbox ~]$ crontab -l
15 * * * * /usr/bin/ps -aux > /home/baatochan/processes_`date +%H-%M-%S`_`date +%Y-%m-%d`.log 2>&1
[baatochan@baatochan-virtualbox ~]$ crontab -r
[baatochan@baatochan-virtualbox ~]$ crontab -l
no crontab for baatochan
[baatochan@baatochan-virtualbox ~]$
```

Sprawdzić czy usługa ssh jest aktywna. Jeśli nie, zainstaluj/uruchom w systemie usługę ssh.

```
[baatochan@baatochan-virtualbox ~]$ systemctl status sshd
● sshd.service - OpenSSH Daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; disabled; vendor preset: disabled)
   Active: inactive (dead)
[baatochan@baatochan-virtualbox ~]$ systemctl start sshd
[baatochan@baatochan-virtualbox ~]$ systemctl status sshd
● sshd.service - OpenSSH Daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; disabled; vendor preset: disabled)
   Active: active (running) since Sat 2020-12-12 18:20:59 CET; 1s ago
     Main PID: 2723 (sshd)
       Tasks: 1 (limit: 4698)
      Memory: 1.7M
      CGroup: /system.slice/sshd.service
             └─2723 sshd: /usr/bin/sshd -D [listener] 0 of 10-100 startups

Dec 12 18:20:59 baatochan-virtualbox systemd[1]: Started OpenSSH Daemon.
Dec 12 18:21:00 baatochan-virtualbox sshd[2723]: Server listening on 0.0.0.0 port 22.
Dec 12 18:21:00 baatochan-virtualbox sshd[2723]: Server listening on :: port 22.
[baatochan@baatochan-virtualbox ~]$
```

Wykonaj kopię pliku konfiguracyjnego usługi ssh.

Plik konfiguracyjny serwera ssh to `/etc/ssh/sshd_config`. Poniżej widać wykonanie kopii domyślnych ustawień serwera ssh. Do dostępu do katalogu i plików ustawień ssh konieczne są uprawnienia roota.

```
[baatochan@baatochan-virtualbox ~]$ sudo cp /etc/ssh/sshd_config /etc/ssh/sshd_config.bac
[sudo] password for baatochan:
[baatochan@baatochan-virtualbox ~]$
```

Wprowadź zmiany w konfiguracji usługi ssh

Ustaw następujące polecenia powitalne "<Imię> - laboratorium 2.1 - <termin zajęć>"

Ustawienie wiadomości powitalnej dzieli się na kilka kroków. Najpierw konieczne jest stworzenie pliku zawierającego odpowiednią treść. Poniżej widać stworzenie pliku `/etc/ssh/welcome.msg`.

```
[baatochan@baatochan-virtualbox ssh]$ sudo nano welcome.msg
[sudo] password for baatochan:
[baatochan@baatochan-virtualbox ssh]$ cat welcome.msg
Bartosz Rodziewicz - laboratorium 2.1 - 02.12.2020
[baatochan@baatochan-virtualbox ssh]$
```

Następnie w konfiguracji ssh należy znaleźć i odkomentować ustawienie `Banner` oraz podać ścieżkę do pliku z wiadomością. Do edycji configu wymagane są uprawnienia roota.

```
#Compression delayed
#ClientAliveInterval 0
#ClientAliveCountMax 3
#UseDNS no
#PidFile /run/sshd.pid
#MaxStartups 10:30:100
#PermitTunnel no
#ChrootDirectory none
#VersionAddendum none

# no default banner path
Banner /etc/ssh/welcome.msg

# override default of no subsystems
Subsystem sftp /usr/lib/ssh/sftp-server

# Example of overriding settings on a per-user basis
#Match User anoncvs
#    X11Forwarding no
#    AllowTcpForwarding no
#    PermitTTY no
#    ForceCommand cvs server
```



Zmień domyślny port, na którym pracuje usługa ssh (jaki?) na inny (spoza puli well-known ports)

Serwer ssh domyślnie pracuje na porcie 22. Zmiana tego portu wymaga odkomentowania `Port` w ustawieniach serwera i zmianie wartości na oczekiwany port.

```
# $OpenBSD: sshd_config,v 1.103 2018/04/09 20:41:22 tj Exp $

# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/sbin:/usr/local/bin:/usr/bin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options override the
# default value.

Port 2137
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
```

Kolejnym krokiem jest restart usługi ssh komendą `systemctl restart sshd`.

Po restarcie usługi ssh połączyć się z serwerem lokalnie

Do połączenia poza adresem ( `localhost` ) należy podać parametr `-p`.

```
[baatochan@baatochan-virtualbox ~]$ ssh localhost
ssh: connect to host localhost port 22: Connection refused
[baatochan@baatochan-virtualbox ~]$ ssh localhost -p 2137
Bartosz Rodziewicz - laboratorium 2.1 - 02.12.2020
baatochan@localhost's password:
Last login: Sat Dec 12 18:35:56 2020 from ::1
[baatochan@baatochan-virtualbox ~]$ exit
logout
Connection to localhost closed.
[baatochan@baatochan-virtualbox ~]$
```

Spróbować połączyć się z innej maszyny (np. z poziomu systemu Windows - klient Putty) w celu zweryfikowania dokonanych

zmian. Do połączenia z hosta konieczna była zmiana ustawień maszyny wirtualnej, aby karta sieciowa była w trybie bridged. Adres IP maszyny wirtualnej można po tym odczytać komendą `ifconfig -a`. Połączenie z hosta zostało wykonane z Windowsa z użyciem powershella, ponieważ nie posiadam klienta putty.

```
PS C:\Users\barto> ssh baatochan@192.168.1.58 -p 2137
Bartosz Rodziewicz - laboratorium 2.1 - 02.12.2020
baatochan@192.168.1.58's password:
Last login: Sat Dec 12 19:00:20 2020 from 192.168.1.43
[baatochan@baatochan-virtualbox ~]$ exit
logout
Connection to 192.168.1.58 closed.
PS C:\Users\barto>
```

Przywrócić oryginalny plik konfiguracyjny usługi ssh

```
[baatochan@baatochan-virtualbox ~]$ sudo cp /etc/ssh/sshd_config.bac /etc/ssh/sshd_config
[sudo] password for baatochan:
[baatochan@baatochan-virtualbox ~]$ sudo systemctl restart sshd
[baatochan@baatochan-virtualbox ~]$ ssh localhost
baatochan@localhost's password:
Last login: Sat Dec 12 19:10:20 2020 from ::1
[baatochan@baatochan-virtualbox ~]$ exit
logout
Connection to localhost closed.
[baatochan@baatochan-virtualbox ~]$
```