

Kurs administrowania systemem Linux 2024

Lista zadań na pracownię nr 5

Na zajęcia 25 i 27 marca 2024

Zadanie 1 (1 pkt). Zademonstruj różnicę pomiędzy współdzieleniem przez dwa procesy deskryptora pliku i dwukrotnym otwarciem tego samego pliku w dwóch procesach. Przygotuj w tym celu skrypt, który w pętli czyta po jednym wierszu ze swojego standardowego wejścia i wypisuje go na standardowe wyjście. Po wypisaniu wiersza śpi podaną jako argument liczbę sekund. Np. wywołanie

```
$ ./czytaj 1.25 < wielowierszowy.txt
```

spowoduje wypisywanie na terminal kolejnych wierszy pliku `wielowierszowy.txt` co 1.25s. Uruchom następnie w tle dwie instancje tego polecenia z nieco różniącymi się czasami (np. 1s i 1.6s) i wyjaśnij, co pojawiło się na ekranie. Uruchom następnie w tle jeszcze raz te polecenia z tymi samymi czasami, ale przekaż obu instancjom ten sam deskryptor otwartego pliku `wielowierszowy.txt`. Wyjaśnij, co teraz się stało.

Powtórz następnie eksperyment z zapisywaniem do pliku oraz z zapisywaniem do pliku w trybie dołączania (najlepiej wierszy różnej długości). Sformułuj wnioski, jak zorganizować zapisywanie danych do wspólnego pliku przez wiele niezależnych procesów tak, aby procesy wzajemnie nie uszkadzały sobie zapisywanych wierszy.

Zadanie 2 (1 pkt). Zapoznaj się z opcjami polecenia `ls(1)` oraz poleceniami `stat(1)`, `realpath(1)`, `readlink(1)`, `dirname(1)` i `basename(1)`. Przygotuj krótkie omówienie sposobów odczytywania informacji o plikach i katalogach. Wyjaśnij, co robi poniższe polecenie:

```
for IMG in *.jpg
do
    convert $IMG $(basename $IMG .jpg).png
done
```

Zadanie 3 (1 pkt). Zapoznaj się z plikami

```
hostname(5), machine-id(5), os-release(5)
```

(jeśli używasz Debiana, to także `/etc/debian_version`) oraz poleceniami

```
hostname(1), hostnamectl(1), uuid(1), uuidgen(1),
dbus-uuidgen(1), uname(1), lsb_release(1).
```

Przygotuj krótkie omówienie sposobów identyfikacji dystrybucji Linuksa oraz konkretnej instalacji.

Zadanie 4 (1 pkt). Zapoznaj się z poleceniami `lsof(8)` i `fuser(1)`. Przygotuj krótkie omówienie zakresu ich użycia i różnic pomiędzy nimi. Jak ujawnić wszystkie pliki otwarte przez podany proces? Jak sprawdzić, który proces otworzył dany plik? Jak ujawnić wszystkie pliki otwarte przez podanego użytkownika?

Zadanie 5 (1 pkt). Zapoznaj się z poleceniem `strace(1)`. Przygotuj krótkie omówienie podstawowych scenariuszy użycia tego programu.

Zadanie 6 (1 pkt). Zapoznaj się z poleceniem `mkfifo(1)`. Napisz skrypt `mylogs`, który tworzy rurę `/tmp/mylog.fifo`, odczytuje z niej wiadomości i wysyła je na standardowe wyjście, dodając czas otrzymania wiadomości w formacie `yyy-mm-dd hh:mm:ss`. Napisz kilka skryptów, które zapisują do rury `/tmp/mylog.fifo` swoje komunikaty diagnostyczne.

Zadanie 7 (1 pkt). Zapoznaj się z programami `netstat(8)` i `ss(8)` oraz z opcją `-U` polecenia `lsof(8)`. Przygotuj krótkie omówienie sposobów uzyskiwania informacji o czynnych gniazdach. Jak się dowiedzieć, który proces utworzył gniazdo? Jak się dowiedzieć, które gniazda należą do podanego procesu?

Zadanie 8 (1 pkt). Zapoznaj się z programami `socat(1)` i `nc(1)` (w wersji *Traditional* i *OpenBSD*). Przygotuj krótkie omówienie ich zastosowań i różnic pomiędzy nimi. Pokaż, jak podejrzeć komunikację sieciową pomiędzy przeglądarką i serwerem WWW. Napisz skrypt, który utworzy lokalne gniazdo nazwane `/tmp/mystat.socket` i będzie odpowiadał na zapytania o bieżący czas, obciążenie systemu itp.

Zadanie 9 (1 pkt). Dowiedz się, co to są przekierowania sieciowe w bashu i przygotuj ich krótkie omówienie. Zadeemonstruj ich działanie wykonując odpowiednio przygotowany przykładowy skrypt.

Zadanie 10 (1 pkt). Dowiedz się, co to jest DBus. Użyj polecenia `dbus-monitor(1)` w celu obserwacji komunikacji na tej magistrali. Wygeneruj różne zdarzenia w systemie (odłącz zasilacz sieciowy itp.) i pokaż, jakie komunikaty przesyła DBus swoim klientom.

Zadanie 11 (1 pkt). Mój mały laptop ma ekran o przekątnej 11.6" i rozdzielczości 1366×768 . Dla wygodnej pracy odległość ekranu od oczu wynosi około 60 cm. Przy mojej ostrości wzroku dobrze widzę czcionki o rozmiarze 12×24 px, tj. 2.25×4.5 mm, co odpowiada rozmiarom kątowym rzędu $12' \times 24'$ (pojedynczy piksel ma przy podanej odległości szerokość około 1 minuty kątowej, co zwykle przyjmuje się za granicę rozdzielczości przeciętnego ludzkiego oka). Daję radę odczytać czcionkę o rozmiarze 8×16 px (standardowa czcionka konsolowa), ale dłuższa praca powoduje zmęczenie wzroku i spadek mojej wydajności.

Mimo iż obraz na ekranie rzutnika wydaje się bardzo duży, to okazuje się, że rozmiary kątowe obrazu z rzutnika są znacznie mniejsze, niż ekranu laptopa podczas normalnej pracy. Obraz z rzutnika ma szerokość 1.5 m, ale jest oglądany z odległości rzędu 5 metrów. Moja ulubiona czcionka do normalnej pracy ma wówczas rozmiary kątowe czcionki 8×16 px oglądanej z odległości 60 cm — daje się odczytać, ale jest to dosyć męczące. Aby osiągnąć na obrazie z rzutnika rozmiary kątowe rzędu $12' \times 24'$, potrzebuję powiększyć czcionkę do rozmiaru 16×32 px. Pełny ekran mieści wówczas 24 wiersze po 85 znaków (czyli nieznacznie więcej, niż standardowy terminal 80×24).

Biorąc pod uwagę wymienione wyżej tryby użycia skonfigurowałem program URxvt tak, by wyświetlał konsolę w trybie pełnoekranowym i by za pomocą klawiszy `<ctrl>--` i `<ctrl>++` można było przełączać pomiędzy rozmiarami 12×24 i 16×32 pikseli.

Zapoznaj się z emulatorami terminali XTerm i URxvt oraz sposobami ich konfigurowania (w tym z plikami `~/.Xresources` i `~/.Xdefaults` oraz poleceniem `xrdb(1)`), a także z poleceniem `xrandr(1)` i przygotuj odpowiedni skrypt, za pomocą którego będziesz mógł szybko i wygodnie przygotować laptop do prezentacji swoich programów na ekranie rzutnika. Przejrzyj dodatkowo dostępną ofertę i wybierz estetyczną i czytelną czcionkę o stałej szerokości. Skonfiguruj także ładną paletę kolorów.