

# Kurs administrowania systemem Linux 2021

Lista zadań na pracownię nr 4

Na zajęcia 23 i 24 marca 2021

**Zadanie 1 (1 pkt).** Zapoznaj się z opcjami polecenia `ls(1)` oraz poleceniami `stat(1)`, `realpath(1)` i `readlink(1)`. Przygotuj krótkie omówienie sposobów odczytywania informacji o plikach i katalogach.

**Zadanie 2 (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 3 (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 4 (1 pkt).** Zapoznaj się z poleceniem `strace(1)`. Przygotuj krótkie omówienie podstawowych scenariuszy użycia tego programu.

**Zadanie 5 (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 6 (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 7 (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 8 (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 9 (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 10 (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' × 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  $8 \times 16$  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  $8 \times 16$  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  $16 \times 32$  px. Pełny ekran mieści wówczas 24 wiersze po 85 znaków (czyli nieznacznie więcej, niż standardowy terminal  $80 \times 24$ ).

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  $12 \times 24$  i  $16 \times 32$  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.<sup>1</sup> Przejrzyj dodatkowo dostępną ofertę i wybierz estetyczną i czytelną czcionkę o stałej szerokości. Skonfiguruj także ładną paletę kolorów.

---

<sup>1</sup>Zadanie wymyśliłem na zajęcia w sali ćwiczeniowej, ale pozostaje aktualne również podczas zajęć zdalnych — obraz udostępniany w MS Teams jest mniejszy, niż widoczny na oryginalnym ekranie.