

Kurs administrowania systemem Linux 2025

Lista zadań na pracownię nr 2: Powtórka z PWI

Na zajęcia 5 i 6 marca 2025

Zadanie 1 (2+1 pkt). Zajrzyj do serwisu `kernel.org`. Skopiuj z niego archiwa `tar.xz` wydań jądra Linuksa aktualnych 1 stycznia w kolejnych nieparzystych latach (tj. 0.99.15j w 1993 roku, 1.1.95 w 1995 roku itd., razem 16 wydań). Zapoznaj się z narzędziami `cloc` i `sloccount` i przygotuj ich krótkie omówienie. Opracuj statystyki liczby wierszy kodu, liczby plików i kosztu odtworzeniowego (liczba osobolat i koszt w dolarach) dla tych 16 momentów życia Linuksa (2 pkt). W miarę możliwości spróbuj zautomatyzować swoją pracę. Program `gunplot` lub Pythonowa biblioteka `matplotlib` pozwolą ładnie zobrazować wyniki (dodatkowy punkt za wizualizację).

Zadanie 2 (1 pkt). Dodaj do pliku `~/.bashrc` definicje (w postaci aliasów bądź funkcji) kilku prostych, ale często przydatnych poleceń:

- `ll` — wypisuje zawartość (podanego lub bieżącego) katalogu korzystając z polecenia `ls` z opcjami `-lAFbhv` oraz `--color=always` (wyjaśnij, co znaczą te opcje) i w celu stronicowania listingu korzysta z `pager less` z opcjami `-XFR` (wyjaśnij, co one znaczą).
- `gentmp` — wypisuje na standardowe wyjście napis `tmp-YYYYMMDDhhmmss` zawierający w miejscu odpowiednich liter bieżący czas.
- `genpwd` — wypisuje na standardowe wyjście napis złożony z 32 losowo wybranych znaków ze zbioru `[3-9A-HJ-NP-Z]` (pomijamy znaki, które można pomylić z innymi, np. `00` itp.), zakończony znakiem nowego wiersza. Jako źródła losowości użyj `/dev/urandom`. Wszystkie ciągi powinny być jednakowo prawdopodobne.

Zadanie 3 (1 pkt). Zapoznaj się z dokumentacją programu `grep(1)` i przygotuj krótkie omówienie wyrażeń regularnych oraz podstawowych opcji programu.

Zadanie 4 (1 pkt). Zapoznaj się z dokumentacją programu `find(1)` i przygotuj krótkie omówienie jego podstawowych opcji.

Zadanie 5 (1 pkt). Zapoznaj się z dokumentacją perlowego skryptu `rename` i przygotuj krótkie omówienie jego użycia. Uwaga: istnieją inne programy (nie napisane w Perlu) o tej samej nazwie.

Zadanie 6 (1 pkt). Podczas instalacji większości pakietów w Debianie/Ubuntu¹ jest tworzony podkatalog katalogu `/usr/share/doc/` o takiej samej nazwie, jak nazwa pakietu, zawierający podstawową dokumentację pakietu. W większości przypadków zawiera on plik `changelog.Debian.gz`, zawierający chronologiczną listę zmian dokonanych w tym pakiecie. Napisz odpowiednie polecenia, które wypiszą następujące informacje:

- Listę zainstalowanych pakietów, które nie posiadają własnego podkatalogu w `/usr/share/doc/`.
- Listę podkatalogów katalogu `/usr/share/doc/`, których nazwy nie są nazwami żadnego zainstalowanego pakietu. Przy każdym z takich podkatalogów wypisz nazwę pakietu, który jest jego właścicielem.
- Listę pakietów posiadających własny podkatalog w katalogu `/usr/share/doc/`, który jednak nie zawiera pliku `changelog.Debian.gz`.

¹Jeśli używasz np. Archa, to dostosuj treść zadania do specyfiki Twojego systemu.

- Listę pakietów posiadających własny plik `changelog.Debian.gz`, który zawiera tylko jeden wpis (zwykle *Initial release*).
- Liczbę wystąpień słowa `bash` (zapisanego małymi lub wielkimi literami) w pliku

`/usr/share/doc/bash/INTRO.gz`

Zadanie 7 (1 pkt). *Biblioteka współdzielona* to plik, którego nazwa pasuje do wzorca

`.*\s\(\.[0-9]\+\)\?&`

Napisz odpowiednie polecenia, które wypiszą następujące informacje:

- Listę wszystkich bibliotek współdzielonych zainstalowanych w Twoim systemie.
- Listę dowiązań symbolicznych do bibliotek współdzielonych zainstalowanych w Twoim systemie.
- Liczbę, sumaryczny rozmiar w bajtach i średni rozmiar wszystkich bibliotek współdzielonych zainstalowanych w Twoim systemie.
- Listę (uporządkowaną i bez powtórzeń) wszystkich katalogów, w których występują biblioteki współdzielone.
- Listę (uporządkowaną i bez powtórzeń) wszystkich katalogów, w których występują dowiązania symboliczne do bibliotek współdzielonych zainstalowanych w Twoim systemie.

Zadanie 8 (1 pkt). Dane podsystemu obsługującego komunikaty w językach narodowych znajdują się w katalogu `/usr/share/locale/`. Zawiera on podkatalogi o nazwach będących nazwami języków w kodowaniu ISO 639-1. Zawierają one m. in. podkatalog `LC_MESSAGES` zawierający pliki `MO` zawierające tłumaczenia komunikatów programów (dane dla biblioteki GNU Gettext wykorzystywanej przez te programy). Napisz odpowiednie polecenia, które wypiszą następujące informacje:

- Listę wszystkich nazw języków, dla których istnieje plik `MO` co najmniej jednego programu.
- Listę wszystkich nazw języków, dla których istnieją komunikaty programu `dpkg`.
- Listę wszystkich programów posiadających komunikaty w języku `pl`.

Strony podręcznika systemowego `man(1)` w różnych językach znajdują się w katalogu `/usr/share/man`. Tradycyjnie znajdowały się tam tylko podkatalogi `man1–man8` odpowiadające ośmiu rozdziałom podręcznika. Obecnie znajdują się tam również katalogi o nazwach będących nazwami języków w kodowaniu ISO 639-1 zawierające podkatalogi `man1–man8`. Podkatalogi te zawierają różnojęzyczne strony podręcznika zawarte w plikach o nazwach pasujących do `globa *.[1-8]`. Napisz odpowiednie polecenia, które wypiszą następujące informacje:

- Dla każdego z ośmiu rozdziałów podręcznika listę wszystkich nazw języków, dla których istnieje co najmniej jedna strona dokumentacji w danym języku i w danym rozdziale. Pamiętaj że zamiast katalogu `en/man1` mamy katalog `man1/` itd.
- Dla każdego z ośmiu rozdziałów podręcznika listę wszystkich stron podręcznika w języku `pl`.

Zadanie 9 (1 pkt). Aby szybko wyszukiwać wiele plików w drzewie katalogów możemy zamiast programu `find` użyć następującego sposobu: za pomocą polecenia `ls` z odpowiednimi opcjami zapisujemy ścieżki do wszystkich plików z pewnego drzewa katalogów w pliku tekstowym (zwykle o nazwie `ls-R`). Aby wyszukać plik przeszukujemy plik `ls-R` za pomocą programu `grep`. Zaprogramuj w Bashu proste narzędzie, które pozwoli wygenerować taki plik `ls-R`, a następnie w nim wyszukiwać. Zapoznaj się następnie z poleceniami `mlocate(1)` i `updatedb(8)` (czemu są lepsze niż stare dobre `locate`?) oraz z `TEX`-ową biblioteką `KPathSea` (w tym z poleceniem `kpsewhich(1)`) i przygotuj ich omówienie.

Zadanie 10 (1 pkt). Zapoznaj się ze stroną podręcznika programu `sed(1)` i przygotuj jego krótkie omówienie.

Zadanie 11 (1 pkt). *Zdanie*, to ciąg znaków zakończony kropką, po której następuje biały znak. Napisz skrypt w języku `sed`, który wczytuje ciąg zdań i wypisuje po jednym zdaniu w wierszu.² Np. dla strumienia wejściowego:

```
To jest
pierwsze
    zdanie. A to już
drugie
    zdanie, które
    jest nieco
dłuższe.
```

program wyprodukuje strumień:

```
To jest pierwsze zdanie.
A to już drugie zdanie, które jest nieco dłuższe.
```

Zadanie 12 (1 pkt). Oto jeden z popularnych przykładów rozważanych w Teorii Przepisywania: n kameleonów siedzi w rzędzie. Każdy z nich jest w danej chwili czerwony **R**, zielony **G** bądź niebieski **B**. Kameleony chciałyby być niewidoczne, dlatego dwa sąsiadujące kameleony różnych kolorów zmieniają kolor na trzeci. Zmiany może zawsze dokonać pierwsza od lewej para różnokolorowych kameleonów, zatem np. ciąg `RRRRGBGB` zamieni się na `RRRBBBGB`, a ten na `RRGGBBGB` itd. Napisz skrypt w języku `sed`, który zobrazuje krok po kroku zmiany ciągu kameleonów. Uwaga: nie zawsze kameleony osiągną monochromatyczność. Czasem ciąg zmian, a zatem i nasz skrypt, może się zapętlić!

Zadanie 13 (1 pkt). Napisz w `sed`-ie symulator maszyny Turinga.

²Pisanie tekstów w języku naturalnym w taki sposób, że każde zdanie jest umieszczone w osobnym wierszu jest wygodne, jeśli chcemy porównywać wersje tekstu za pomocą programu `diff` albo polecenia `git diff`. Wiele osób składających teksty za pomocą systemu `TEX` stosuje tę konwencję.