

Technologie Informacyjne i Komunikacyjne R Linux

- Zad. 1.** Rozpocznij pracę w środowisku graficznym systemu Linux.
- Zad. 2.** Przećwicz przełączanie pomiędzy trybem graficznym a trybem tekstowym.
- Zad. 3.** Zapoznaj się z programami `ls`, `rm`, `mv`... za pomocą programu `man`.
- Zad. 4.** Do ścieżki przeszukiwania określonej przez zmienną środowiskową `PATH` dopisz na początku katalog bieżący `.` (kropka). Wynik operacji sprawdź poleceniem `printenv` lub `echo`.
- Zad. 5.** Wyświetl zawartość katalogu domowego (korzystając z różnych opcji programu `ls`).
- Zad. 6.** Korzystając z polecenia `cd` przejdź do swojego katalogu roboczego (`_work_`) i sprawdź, gdzie znajdujesz się LOGICZNIE i FIZYCZNIE. Następnie wróć do katalogu domowego (w tym samym terminalu!).
- Zad. 7.** Poleceniem `alias` sprawdź istniejące aliasy, a następnie zdefiniuj:
- ```
alias cp="cp -i"
alias mv="mv -i"
alias rm="rm -i"
alias la="ls -la"
```
- Sprawdź ich działanie.
- Zad. 8.** Usuń alias zdefiniowany na polecenie `cp`, a pozostałe dopisz do pliku `.bashrc` w katalogu domowym użytkownika (aby polecenia wykonywały się automatycznie po otwarciu nowego terminala). Na końcu pliku wciśnij ENTER. Otwórz nowy terminal i sprawdź, czy nowe aliasy są zdefiniowane.
- Zad. 9.** Wyświetl zawartość katalogu nadrzędnego. Następnie wyświetl tylko katalogi i pliki zaczynające się na `m`. Po czym wyświetl tylko elementy, których ostatnią cyfrą nazwy jest `3`.
- Zad. 10.** Przy pomocy operacji przekierowania strumienia wejściowego wczytaj zawartość katalogu nadrzędnego do pliku `lista`
- Zad. 11.** Sprawdź jakiego typu jest plik i nadaj mu odpowiednie rozszerzenie.

- Zad. 12.** Wyświetl zawartość pliku za pomocą programów do wyświetlania i edycji plików tekstowych: `more`, `less`, `cat`, `nano`, `gedit`, `vi`... (dowolnie).
- Zad. 13.** Sprawdź ile linii ma plik `lista.txt`
- Zad. 14.** Wyświetl nagłówek i stopkę pliku `lista.txt`, następnie pierwsze i ostatnie 15 linii tego pliku.
- Zad. 15.** Utwórz katalog `cw1`, przekopiuj do niego plik `lista.txt`. Usuń plik `lista.txt` z katalogu domowego.
- Zad. 16.** Za pomocą polecenia `sort`, posortuj **losowo** zawartość pliku `lista` i wynik działania przekieruj do pliku `random.txt`.
- Zad. 17.** Korzystając z programu `find` znajdź wszystkie pliki o nazwie `README` znajdujące się w części drzewa katalogowego rozpoczynającej się od katalogu `/usr`. Wynik zapisz w pliku `/cw1/find.out`.
- Zad. 18\*.** Korzystając z programu `sed` na pliku `find.out` zamień ścieżki do katalogu `/usr/` na ścieżkę do swojego katalogu domowego.
- Zad. 19.** Poleceniem `umask` odczytaj maksymalne uprawnienia dla nowotworzonego pliku, a następnie ogranicz je a taki sposób, aby odczytywać i tworzyć plik mógł wyłącznie właściciel pliku.
- Zad. 20.** Utwórz pusty plik `info.txt` w katalogu `cw1` i sprawdź, czy przypisano mu uprawnienia zdefiniowane w zadaniu powyżej.
- Zad. 21.** Korzystając z edytora tekstu `nano` napisz dowolny nagłówek w pliku `info.txt` (np. „Informacje o procesorze”). Następnie korzystając z polecenia przekierowania strumienia wejściowego **dopisz** do tego pliku informacje techniczne o Twoim komputerze, które znajdują się w pliku `/proc/cpuinfo/`.
- Zad. 22.** Plikowi `info.txt` nadaj takie uprawnienia dostępu, aby mógł być wyświetlany lecz nie modyfikowany przez Twojego sąsiada. Natomiast Twojemu użytkownikowi nadaj uprawnienia do tworzenia, ale nie do czytania tego pliku. Spróbuj wyświetlić plik `info.txt`.
- Zad. 23.** W katalogu domowym utwórz archiwum, kompresując katalog `cw1` wraz z zawartością.
- Zad. 24.** Za pomocą programów: archiwizującego `tar` i programu zliczającego znaki `wc` oraz z wykorzystaniem przesłania wyniku działania polecenia na wejście kolejnego polecenia (tzw. pipe), zlicz liczbę plików w utworzonym archiwum.

**Zad. 25.** Posługując się programami `find` i `wc` znajdź liczbę plików o rozmiarze większym od 1,000,000 bajtów znajdujących się w części drzewa katalogowego rozpoczynającej się od katalogu `/usr`.

**Zad. 26.** Przy pomocy programów `tar` i `wc` odpowiedz na pytanie jaki będzie rozmiar archiwum (bez kompresji) zawierającego cały Twój katalog domowy. Następnie sprawdź ile miejsca będą wymagały te dane po kompresji programem `gzip`.

**Zad. 27.** Korzystając z poleceń `du`, `sort` i `tail` wskaż w swoim katalogu domowym podkatalog, który zajmuje najwięcej i najmniej miejsca.

**Zad. 28.** Polecenie `ypcat passwd` wypisuje na ekranie bazę danych użytkowników Ośrodka Komputerowego Wydziału Fizyki.

- Posługując się programami `grep` i `wc` dowiedz się ilu użytkowników ma na imię tak samo jak Ty i ile osób jest na Twoim roku studiów.
- Wypisz same imiona i nazwiska tych osób do pliku o nazwie `nazwiska.txt` (wskazówka: można skorzystać z polecenia `cut`).

**Zad. 29.** Korzystając z polecenia `ps` z opcjami `-ef` pokazującego listę wszystkich procesów uruchomionych na komputerze, zapisz do pliku `procesy.txt` tylko własne procesy. Spróbuj wykonać to zadanie korzystając z polecenia `ps` tylko z opcjami `-ef` i polecenia `grep`. Następnie, znajdź proces-rodzica dla swojego terminala i zamknij okno terminala usuwając stosowny proces przy pomocy polecenia `kill`.

**Zad. 30.** Za pomocą polecenia `ssh` połącz się z komputerem `primus.okwf.fuw.edu.pl` i sprawdź, czy masz dostęp do tych samych plików co w pracowni komputerowej i czy nowe aliasy są automatycznie zdefiniowane. Rozłącz się z komputerem za pomocą polecenia `logout`.

**Zad. 31.** Z poziomu terminala uruchom przeglądarkę WWW, a następnie nie kończąc pracy programu, przywróć możliwość wydawania kolejnych poleceń w tym terminalu, odsuwając działanie przeglądarki w tryb pracy w tle.

**Zad. 32.** W katalogu roboczym `_work_` utwórz katalog `test`. Następnie w katalogu `pracownia1` utwórz katalog-link do katalogu `test`, a ścieżkę do tego katalogu wyeksportuj do nowej zmiennej.

**Zad. 33\*.** Upewnij się, że alias `rm='rm -i'` jest aktywny. W katalogu `test` utwórz 1000 pustych plików z rozszerzeniem `.dat`, korzystając z pętli

for i polecenia wyświetlającego sekwencję liczb `seq`. Następnie, posiłkując się programem `yes`, wypisującym literę `y` w terminalu, usuń wszystkie pliki.

**Zad. 34.** Napisz skrypt `first.sh`, który wyświetli na ekranie “Hello world!” (ważne aby pamiętać o przejściu do nowego wiersza w ostatnim wierszu).

**Zad. 35.** Korzystając ze **zmiennych środowiska** stwórz skrypt, który po uruchomieniu będzie wypisywał na ekran następujące komunikaty:

```
Witaj KKK
Pracujesz na komputerze LLL
Używasz systemu MMM
Używasz powłoki NNN
Twój katalog domowy to OOO
```

Symbole KKK, LLL, MMM, NNN, OOO powinny automatycznie zostać zastąpione odpowiednio przez: nazwę użytkownika, nazwę komputera, nazwę systemu operacyjnego, nazwę używanej powłoki i nazwę katalogu domowego.

**Zad. 36.** Napisz skrypt, który doda do siebie dwie liczby podane jako dwa kolejne argumenty wejściowe skryptu, a wynik wyświetli w terminalu. Następnie zmodyfikuj ten skrypt tak, aby argumenty były wczytywane po uruchomieniu skryptu.

**Zad. 37\*.** Napisz skrypt–grę–generator liczb pseudolosowych, w której użytkownik podaje liczbę (w czasie działania skryptu) w przedziale od 1 do 100. Użytkownik wygrywa jeśli poda liczbę większą, niż liczba wylosowana przez generator. Skrypt powinien wyświetlić wynik gry, liczby: podaną przez użytkownika i wylosowaną przez generator. *Wskazówka:* Liczba pseudolosowa nie powinna zmieniać wartości podczas działania skryptu.

**Zad. 38.** Napisz skrypt, który stworzy katalog o nazwie podanej jako pierwszy argument wywołania skryptu, a następnie skopiuje z katalogu podanego jako drugi argument wywołania skryptu wszystkie pliki o rozszerzeniach podanych (bez kropki) jako trzeci argument wywołania skryptu. W skrypcie:

- powinno być sprawdzane czy podano wszystkie niezbędne argumenty i jeśli nie, należy wyświetlić odpowiedni komunikat;
- powinno być sprawdzane czy katalog docelowy istnieje i jeżeli nie istnieje to trzeba go utworzyć, a jeśli istnieje, to wypisać komunikat że podano nazwę istniejącego katalogu.

**Zad. 39.** Napisz skrypt, który będzie wyświetlał z bieżącego katalogu nazwy wszystkich plików o rozszerzeniach `*.txt`. Skorzystaj z pętli `for`.

**Zad. 40\*.** Napisz skrypt startujący i kończący działanie aplikacji. Skrypt powinien przyjmować 2 argumenty wejściowe: polecenie `start` lub `stop` i nazwę aplikacji, którą ma obsługiwać. Skrypt powinien sprawdzić poprawność poleceń wejściowych, a jeżeli polecenia nie są poprawne, wyświetlić odpowiedni komunikat. Aplikacja powinna zostać uruchomiona w tle. *Wskazówki:* Warto skorzystać z instrukcji warunkowej `case`, poleceń `ps`, `grep`, `cut` i `kill` z odpowiednimi opcjami. Należy uważnie odczytać ID procesu aplikacji. Miłej zabawy!