

# Technologie Informacyjne - Linux 2

Instytut Matematyki  
Uniwersytet Gdański

Polecenie **grep** wypisuje z pliku lub strumienia wejściowego te wiersze, które są zgodne z podanym wyrażeniem.

Na przykład

**\$ grep root /etc/passwd** - wypisze wszystkie wiersze z pliku /etc/passwd, które zawierają w sobie słowo root.

**\$ grep root /etc/\*** - wyszuka wiersze ze słowem root we wszystkich plikach znajdujących się w katalogu /etc.

Opcja **-i** wyłącza rozróżnienia wielkich i małych liter.

Opcja **-v** powoduje wypisywanie wierszy niezgodnych z wyrażeniem.

**Zadanie** Pobrać plik lista.txt a następnie za pomocą polecenia grep wypisać wszystkie osoby:

- a) o imieniu Wojciech,
- b) zapisz wszystkie osoby o imieniu Marta do pliku Marta.doc,
- c) zapisz wszystkie osoby o imieniu innym niż Marta do pliku NieMarta.odt

**Zadanie** Pobierz plik pesel.txt. Za pomocą odpowiedniego polecenia sprawdzić, czy w pliku jest pesel:

- a) 67112966668, b) 73070871369.

Polecenie **less** (w starszych wersjach **more**) służy do przeglądania zawartości dużego pliku. **Spacja** powoduje przesunięcie tekstu o jeden ekran do przodu, naciśnięcie **klawisza B** - przewinięcie o jeden ekran do tyłu. **Klawisz Q** powoduje zakończenie pracy programu.

Wewnątrz polecenia less możliwe jest **poszukiwanie tekstu**. W celu poszukiwania słowa w dalszej części tekstu wpisujemy **/słowo**, w celu poszukiwania tego słowa wstecz wpisujemy **?słowo**.

## Zadanie

- Sprawdź działanie polecenia less na przykładzie pliku pesel.txt
- Wyszukaj pesele wszystkich osób, które urodziły się w październiku 1966 roku.

Możliwe jest przesłanie standardowego wyjścia niemal każdego programu bezpośrednio na standardowe wejście innego programu. Przykład wysłania wyników programu `grep` do polecenia `less`:

```
$ grep ie /usr/share/dic/words | less
```

**Zadanie** Wyświetl za pomocą polecenia `less` wszystkie osoby z listy.txt, które nie mają na imię Wojciech.

Polecenie **pwd** wypisuje nazwę aktualnego katalogu roboczego.

Polecenie **find kat -name plik -print** pozwala znaleźć plik o nazwie plik w katalogu kat.

Polecenie **file plik** pozwala sprawdzić rozszerzenie pliku

## Zadanie

- Wypisz nazwę aktualnego katalogu roboczego.
- Będąc w katalogu głównym znaleźć lokalizacje pliku Marta.txt oraz katalogów Informatyka i Analiza Matematyczna.
- Wyświetlić informacje o rozszerzeniach pliku lista.txt, Marta.doc oraz NieMarta.odt.

Polecenie **diff plik1 plik2** pozwala na wyłuszczenie różnicy pomiędzy dwoma plikami tekstowymi.

Polecenie **head plik** wypisuje pierwszych dziesięć wierszy pliku, a polecenie **tail plik** dziesięć ostatnich wierszy. Liczbę wierszy, którą chcemy wypisać można zmienić stosując opcję **-n**, gdzie n to liczba wierszy.

Przykładowo **head -5 /etc/passwd** wypisze pięć pierwszych wierszy z pliku passwd.

Polecenie **tail +n plik** pozwala zobaczyć wiersze poczynając od wiersza o numerze n.

## Zadanie

- a) Pobierz plik lista1.txt. Znajdź różnicę pomiędzy plikami lista oraz lista1.
- b) Zapisać cztery pierwsze osoby z pliku lista.txt do pliku cztery.txt.



# Zmianianie hasła i powłoki

Do zmiany swojego hasła służy polecenie **passwd**. Pytania ono o poprzednie hasło, a następnie prosi o dwukrotne wprowadzenie nowego.

Do zmiany aktywnej powłoki służy polecenie **chsh** albo jedno z poleceń stosowanych zamiennie, takich jak **ksh** lub **tcsh**.

# Zmienne środowiskowe i powłoki

Powłoka może przechowywać **zmienne tymczasowe**, nazywane **zmiennymi powłoki** (shell variables), przechowujące wartości liczbowe lub ciągi znaków. Aby przypisać wartość zmiennej używamy znaku równości: **nazwa\_zmiennej = wartość\_zmiennej**.

Przykładowo

```
$ STUFF=blabla
```

przypisuje zmiennej STUFF wartość blabla.

Dostęp do wartości zmiennej uzyskujemy wpisując przed nazwą zmiennej znak dolara. Przykładowo możemy użyć polecenia: **echo \$STUFF**.

**Zmienne środowiskowe** (environment variables) są bardzo podobne do zmiennych powłoki, ale nie są związane wyłącznie z powłoką. Wszystkie programy w systemach uniksowych mają dostęp do pamięci zmiennych środowiskowych. Zmienną środowiskową tworzymy tak samo jak zmienną powłoki. Następnie przenosimy je do pamięci zmiennej środowiskowej za pomocą polecenia **export**. Przykładowo

```
$ STUFF=blabla
```

```
$ export STUFF
```

**Zmienna PATH** jest specjalną zmienną środowiskową zawierającą **ścieżkę poleceń**. Ścieżka poleceń jest listą katalogów w systemie, które powłoka będzie przeglądać w czasie próby odnalezienia polecenia.

## Zadanie

- Zapisz w zmiennej IMIE swoje imię, a w zmiennej NAZWISKO swoje nazwisko. Wyświetl swoje imię i nazwisko.
- Zapisz zawartość zmiennej PATH w pliku path.

Można **zmienić wartość zmiennej PATH** tak, aby powłoka wyszukiwała programy także w innych katalogach.

Polecenie **PATH=kat:\$PATH** spowoduje, że powłoka najpierw szuka programu w katalogu kat a dopiero następnie w pozostałych katalogach ze zmiennej PATH.

Polecenie **PATH=\$PATH:kat** spowoduje, że katalog kat będzie przeszukiwany na końcu.

Jeśli jest kilka programów o tej samej nazwie to jest uruchamiany pierwszy napotkany.

Polecenie **man** służy do wywoływania podręcznika opisującego poszczególne polecenia. Przykładowo **man ls** wyświetli informacje o poleceniu **ls**.

Polecenie **man -k słowo\_kluczowe** umożliwia wyszukanie informacji z podręcznika dotyczących określonego słowa kluczowego. Przykładowo **man -k sort**.

Jeśli szukamy konkretnej opcji danego polecenia wystarczy po tym poleceniu wpisać **--help**. Przykładowo **ls --help**.

## Zadanie

a) Wyświetl informacje o poleceniu ls. Poszukaj jakie są możliwe dla tego programu kody wyjścia (Exit status) i co one oznaczają?

b) Wyjaśnić znaczenie ciągu poleceń:

**head /proc/cpuinfo | tr a-z A-Z.**

Poszukaj informacji o poleceniu tr w podręczniku.

# Wejście i wyjście powłoki

Czasem może się zdarzyć, że mimo przekierowania standardowego wyjścia program nadal będzie wypisywał różne komunikaty na ekranie. Oznacza to, że korzysta on ze **standardowego strumienia błędów**.

**Zadanie** Wpisz polecenie `ls /ffffff > f`.

Jeśli chcemy, aby dane generowanie przez program były zapisywane do pliku `f`, a wszystkie komunikaty o błędach do pliku `e` to należy użyć polecenia: `ls /ffffff > f 2> e`.

Numer 2 to identyfikator strumienia, który podlega przekierowaniu. Strumieniem numer 1 jest standardowe wyjście, strumieniem numer 2 – standardowy strumień błędów.

Jeśli chcemy przekierować standardowy strumień błędów w to samo miejsce co wynik działania polecenia używamy przekierowania `>&`. Przykładowo `ls /ffffff > f 2>&1`.

**Zadanie** Wywołać dowolną komendę, której wynikiem będzie błąd. Zapisać informację o błędzie w pliku błąd.



**Procesem** nazywamy działający program. Każdy z procesów działających w systemie ma swój liczbowy identyfikator procesu (ang. process ID - PID).

**ps** - wyświetla wszystkie uruchomione procesy,

**ps x** - wyświetla wszystkie uruchomione procesy przez aktualnego użytkownika,

Polecenie **kill** pozwala zakończyć dany proces (**kill pid**).

Polecenie **kill -STOP pid** zatrzymuje działanie danego procesu. Aby wznowić działanie procesu używamy polecenia **kill -CONT pid**.

**Zadanie** Sprawdzić jakie procesy są aktualnie uruchomione na twoim komputerze.