

[Powrót \(Systemy operacyjne\)](#)

Specjalne rodzaje plików, zmienne interpretera poleceń

ćwiczenia laboratoryjne

Ważne komendy (specjalne rodzaje plików)

- `ln` - tworzy dowiązania między plikami.
- `mesg` - kontroluje dostęp zapisu do terminali przez inne osoby.
- `mkfifo` - tworzy potoki nazwane (FIFO).
- `nc` - przekierowuje dane do gniazd sieciowych i plikowych.
- `stat` - wyświetla status pliku lub systemu plików (m. in. liczbę dowiązań)
- `stty` - zmienia ustawienia terminala.
- `tee` - pobiera dane ze strumienia wejściowego i tworzy dwa strumienie wyjściowe: jeden podłączony do standardowego wyjścia, a drugi do wskazanego pliku.
- `tty` - wypisuje nazwę pliku terminala podłączonego do standardowego wejścia.

Ważne komendy (zmienne interpretera poleceń)

- `declare` - inicjuje zmienną z parametrem:
 - `-a` - typu tablicowego,
 - `-f` - tworzy funkcję,
 - `-i` - typu całkowitego,
 - `-p` - wyświetla zmienne oraz ich atrybuty i wartości,
 - `-r` - tylko do odczytu,
 - `-x` - zmienna środowiskowa (odpowiednik polecenie *export*).
- `export` - inicjuje zmienną środowiskową.
 - z parametrem `-n` usuwa zmienną ze środowiska.
- `readonly` - inicjuje zmienną tylko do odczytu.
- `printenv` - wypisuje listę aktualnie ustawionych zmiennych środowiskowych.
- `set` - wyświetla listę zmiennych.
- `source` - ponownie przeładowuje plik.
- `unset` - usuwa zmienną.

Przypisanie wartości zmiennej

- `zmienna=wartość`
- `let zmienna=wartość` - typu całkowitego.
- `zmienna=(wartość1 wartość2 ... wartośćN)` - typu tablicowego.
- `zmienna=([index1]=wartość1 [index2]=wartość2 ... [indexN]=wartośćN)` - typu tablicowego (przypisanie wartości do konkretnych pozycji).

Odniesienie się do wartości zmiennej: `$zmienna`.

Składnia operacji arytmetycznych: `$((wyrażenie))` lub `${wyrażenie}`.

Można je wykonywać jedynie na liczbach całkowitych.

Operacje na zmiennych typu napisowego

- `napis1napis2` - konkatenacja.
- `${#zmienna}` - długość napisu, którego wartość znajduje się w zmiennej.
- `${zmienna:pozycja}` - wartość napisu od *pozycja* (indeksowany od 0).
- `${zmienna:pozycja1:pozycja2}` - wartość napisu od *pozycja1* do *pozycja2* (indeksowany od 0).

Operacje na zmiennych typu całkowitego

- `wartość1+wartość2` - dodawanie.
- `wartość1-wartość2` - odejmowanie.
- `wartość1*wartość2` - mnożenie.
- `wartość1/wartość2` - dzielenie.
- `wartość1%wartość2` - reszta z dzielenia.
- `0wartość` - liczba w systemie ósemkowym.
- `0xwartość` - liczba w systemie szesnastkowym.
- `podstawa#wartość` - liczba w innym systemie liczbowym.

Operacje na zmiennych typu tablicowego

- `${zmienna[*]}` - wszystkie elementy tablicy.

- `${zmienna[indeks]}` - odniesienie się do wartości znajdującej się w tablicy na pozycji *indeks*.
- `${!zmienna[*]}` - wszystkie indeksy.
- `${#zmienna[*]}` - liczba elementów tablicy.

Wartość wyjścia standardowego polecenia: `$(poLecenie)` lub ``poLecenie``.

Zadania

Zadanie 1. Korzystając z odpowiednich poleceń znajdź w systemie plików plik specjalny swojego terminala.

Sprawdź:

- a. jakiego rodzaju jest plik terminala i jakie ma prawa dostępu (`ls -l`).
- b. czy jest możliwe usunięcie swojego pliku swojego terminala.
- c. co się stanie, gdy przekierujesz wyjście/błąd do swojego pliku terminala ze swojego terminala.
- d. co się stanie, gdy przekierujesz wyjście/błąd do swojego pliku terminala z innego terminala.
- e. co się stanie, gdy przekierujesz wyjście polecenia cat pliku terminala do innego terminala.
- f. co się stanie, gdy przekierujesz wejście z pliku terminala do polecenia cat (w celu zakończenia działania przydatny może być `ctrl+c`).

Stosując odpowiednie polecenie wypróbuj możliwość pisania na kilku terminalach jednocześnie.

Możesz spróbować zmienić ustawienia terminala (pamiętaj, że domyślne ustawienia przywraca polecenie: `stty sane`).

Zadanie 2. Znajdź plik który znajduje się w folderze `/dev/null`

Sprawdź:

- a. jakiego rodzaju jest ten plik i jakie ma prawa dostępu (`ls -l`).
- b. co się stanie, gdy przekierujesz coś do tego pliku.
- c. co się stanie, gdy otworzysz w edytorze ten plik, zmienisz zawartość i zapiszesz.
- d. jaki rozmiar będzie miał plik, gdy go przekopiujesz do swojego katalogu domowego.

Zastanów się jakie zastosowanie plik `/dev/null`.

Zadanie 3. Zapoznaj się z możliwością tworzenia dowiązań twardych, np. możesz utworzyć plik `aaa.txt` i utworzyć do niego dowiązanie twarde `bbb.txt`.

Sprawdź:

- a. jakiego rodzaju jest dowiązanie twarde (`ls -l`).
- b. czy dowiązanie twarde można stworzyć dla pliku i katalogu.
- c. jaka jest liczba dowiązań dla pliku.
- d. czy zmiana zawartości jednego pliku będzie widoczna dla drugiego.
- e. czy zmiana praw dostępu jednego pliku będzie widoczna dla drugiego.
- f. usunięcie jednego pliku usuwa drugi.

Zastanów się jakie zastosowanie ma dowiązanie trwałe.

Zadanie 4. Zapoznaj się z możliwością tworzenia dowiązań symbolicznych, np. możesz utworzyć plik `aaa.txt` i utworzyć do niego dowiązanie symboliczne `bbb.txt`.

Sprawdź:

- a. jakiego rodzaju jest dowiązanie symboliczne (`ls -l`).
- b. czy dowiązanie symboliczne można stworzyć dla pliku i katalogu.
- c. jaka jest liczba dowiązań dla pliku/katalogu.
- d. czy zmiana zawartości jednego pliku/katalogu będzie widoczna dla drugiego.
- e. czy zmiana praw dostępu jednego pliku/katalogu będzie widoczna dla drugiego.
- f. usunięcie jednego pliku/katalogu usuwa drugi.

Zastanów się jakie zastosowanie ma dowiązanie symboliczne.

Zadanie 5. Stwórz potok nazwany o dowolnej nazwie.

Sprawdź:

- a. jakiego rodzaju jest potok nazwany (`ls -l`).
- b. co spowoduje przekierowanie jednego wyjścia do potoku nazwanego, a następnie wyświetlenie jego zawartości w innym terminalu.
- c. co spowoduje przekierowanie dwóch lub więcej wyjść do potoku nazwanego, a następnie wyświetlenie jego zawartości w innym terminalu.
- d. co spowoduje początkowe wyświetlenie zawartości potoku nazwanego, a następnie przekierowanie do niego zawartości.

Zastanów się jakie zastosowanie mają potoki nazwane.

Zadanie 6. Stwórz gniazdo o dowolnej nazwie.

Sprawdź:

- jakiego rodzaju jest potok nazwany (1s -1).
- w jaki sposób nasłuchać i wysłać dane poprzez gniazdo.
- czy można wysłać dane więcej niż jeden raz (np. z innego terminala).

Zastanów się jakie zastosowanie mają gniazda.

Zadanie 7. Przejrzyj oraz postaraj się podać jakie wartości zawierają zmienne środowiskowe: `$HOME`, `$LANG`, `$PATH`, `$PWD`, `$SHELL`, `$TERM`, `$USER`.

Zadanie 8. Zapoznaj się z możliwościami tworzenia zmiennych i zmiennych środowiskowych, np.

- utwórz zmienna o nazwie np. `test` i nadaj jej wartość np. `Ala ma kota`.
- wyświetl zawartość zmiennej `test` na ekranie.
- przypisz nową wartość zmiennej `test`, np.: `Kot ma Ale`, następnie wyświetl tę zawartość i sprawdź czy została zmieniona.
- sprawdź czy zmienna `test` jest zmienną środowiskową, jeżeli nie dodaj ją do środowiska.
- usuń zmienną `test` ze środowiska, a następnie sprawdź czy zmienna `test` nadal znajduje się.
- dodaj ponownie zmienną `test` do środowiska i usuń ją całkowicie, następnie sprawdź, czy na liście zmiennych środowiskowych oraz na liście wszystkich,
- utwórz zmienną i zmienną środowiskową, następnie sprawdź czy zmienna i zmienna środowiskowa istnieje w nowo terminalu.

Zadanie 9. Zapoznaj się z możliwościami typów zmiennych, np.

- przypisz dowolną wartość typu napisowego do zmiennych o nazwie `napis1` oraz `napis2` oraz do zmiennej o nazwie `napisy` przypisz konkatenację wartości zmiennych tych dwóch zmiennych.
- wyświetl jaka jest długość wartości zmiennej `napisy`.
- wyświetl wartość zmiennej `napisy` od 3-go znaku indeksowanego od 0.
- wyświetl wartość zmiennej `zmienna` od 3-go do 7-go znaku indeksowanego od 0.
- wyświetl wartość zmiennej `zmienna` od 3-go do 7-go znaku indeksowanego od 0.
- przypisz dowolną wartość typu całkowitego do zmiennych o nazwie `liczba1` i `liczba2` oraz przypisz im dowolne wartości.
- stwórz zmienne typu całkowitego o nazwach `suma`, `roznica`, `iloczyn`, `iloraz` oraz przypisz im wartości operując na zmiennych `liczba1` i `liczba2`.
- stwórz zmienną typu tablicowego o nazwie `tablica` przechowującą i przypisz jej dowolne wartości.
- wyświetl 2-gi element wartości zmiennej `tablica` indeksowanej od 0.
- wyświetl informacje ile elementów zawiera zmienna `tablica`.
- stwórz zmienne przechowujące liczby w różnych systemach liczbowych, wykonaj na nich operacje arytmetyczne i wyświetl wynik.

Zadanie 10. Zapoznaj się z możliwością uruchomienia plików binarnych w dowolnym miejscu (wykorzystaj plik `~/ .bashrc` i odpowiednie zmienne środowiskowe).

Przykładowo plik [example](#), który da się uruchomić poleceniem `./example`, skonfiguruj tak by się dało uruchomić poleceniem `example`, z dowolnej lokalizacji.