

# Systemy operacyjne

Skrypt do ćwiczeń laboratoryjnych

Cezary Sobaniec

\$Id: sop1-zadania.lyx,v 1.19 2006/09/26 13:37:25 sobaniec Exp \$

# Spis treści

<b>1</b>	<b>Wprowadzenie</b>	<b>2</b>
1.1	Podstawy . . . . .	2
1.2	Pomoc systemowa . . . . .	2
1.3	Interpreter poleceń . . . . .	3
<b>2</b>	<b>System plików</b>	<b>4</b>
2.1	Katalogi i pliki . . . . .	4
2.2	Prawa dostępu . . . . .	4
2.3	Wyszukiwanie plików . . . . .	5
2.4	Dowiązania . . . . .	6
<b>3</b>	<b>Edytory</b>	<b>7</b>
3.1	Edytor <code>vi</code> . . . . .	7
3.2	Edytor <code>emacs</code> . . . . .	7
<b>4</b>	<b>Procesy</b>	<b>9</b>
4.1	Lista procesów . . . . .	9
4.2	Sygnały . . . . .	9
4.3	Priorytety procesów . . . . .	9
4.4	Obsługa procesów w interpreterze poleceń . . . . .	10
<b>5</b>	<b>Potoki</b>	<b>11</b>
5.1	Filtr <code>cat</code> . . . . .	11
5.2	Filtry <code>head</code> , <code>tail</code> . . . . .	11
5.3	Filtr <code>grep</code> . . . . .	11
5.4	Filtry <code>wc</code> , <code>tr</code> . . . . .	12
5.5	Filtry <code>cut</code> , <code>sort</code> , <code>uniq</code> . . . . .	12
<b>6</b>	<b>Interpreter poleceń</b>	<b>13</b>
6.1	Zmienne środowiskowe . . . . .	13
6.2	Aliasy i funkcje . . . . .	13
6.3	Konfiguracja . . . . .	13
<b>7</b>	<b>Programowanie w <code>sh</code></b>	<b>14</b>
7.1	Argumenty . . . . .	14
7.2	Instrukcja warunkowa . . . . .	14
7.3	Pętle <code>for</code> i <code>while</code> . . . . .	14
7.4	Wczytywanie wartości . . . . .	15
7.5	Zadania zaawansowane . . . . .	15

# 1

## Wprowadzenie

Aktualna wersja niniejszego dokumentu jest dostępna w postaci elektronicznej pod adresem:

<http://www.cs.put.poznan.pl/csobaniec/Students/Sop/sop1-zadania.pdf>

### 1.1 Podstawy

1. Logowanie do systemu. Zmiana hasła komendą `passwd`. Wylogowanie z systemu.
2. Konsole wirtualne. Tryb graficzny. Emulator terminala.
3. Wydawanie komend w systemie Unix. Przełączniki i argumenty. Podstawowe komendy: `ls`, `pwd`, `cd`.
4. Hierarchia katalogów: `hier(7)`.

### 1.2 Pomoc systemowa

1. Przeglądarka `more`: `q` — wyjście, `Enter`, `j`, `k` — przewijanie liniami, `Spa`, `Ctrl-f`, `Ctrl-b` — przewijanie stronami, `G` — koniec pliku, `/` — wyszukiwanie tekstu.
2. Polecenie `man`. Pomoc systemowa po polsku. Budowa strony pomocy systemowej. Sekcje pomocy systemowej.
3. Programy pomocnicze: `apropos`, `whatis`, `whereis`.
4. Program `info`.

## 1.3 Interpreter poleceń

1. Praca z interpreterem poleceń: przywoływanie ostatnich komend, uzupełnianie nazw plików.
2. Przerywanie działania poleceń: `Ctrl-C`. Przewijanie zawartości okna terminala (`Shift-PgUp`, `Shift-PgDn`).
3. Nazwy uogólnione. Znaki specjalne: `*`, `?` i `[]`.
  - (a) wyświetl listę jednoznakowych programów
  - (b) wyświetl programy dwuliterowe złożone z liter „a”, „b” i „c”
  - (c) wyświetl programy zawierające przynajmniej jedną wielką literę i jedną cyfrę

# 2

## System plików

### 2.1 Katalogi i pliki

1. Utwórz katalog `abc` w katalogu domowym. Utwórz następnie katalog `xyz` w katalogu `abc`. Utwórz strukturę katalogów `x1/x2/x3` w katalogu bieżącym korzystając z przełącznika `-p` polecenia `mkdir`. Usuń całą strukturę podkatalogów jednym wywołaniem polecenia `rmdir`.
2. Skopiuj wszystkie pliki z katalogu domowego kończące się na `.c` i `.txt` do katalogu `abc`. Przejdź do katalogu `abc/xyz` i skopiuj do niego wszystkie pliki kończące się na `.txt` z katalogu macierzystego. Następnie skopiuj całą strukturę katalogów `abc` tworząc analogiczną strukturę o nazwie `abc2` (kopiowanie rekurencyjne). Zbadaj działanie przełącznika `-i` polecenia `cp` przy ponownej próbie kopiowania takiej struktury (zobacz `man cp!`). Zbadaj również działanie przełącznika `-v`.
3. Usuń wszystkie pliki z katalogu `abc2/xyz`. Usuń całą strukturę katalogów `abc2` wraz z plikami (usuwanie rekurencyjne). Sprawdź działanie przełączników `-i` i `-f` polecenia `rm`. Przełącznik `-f` sprawdź w odniesieniu do plików tylko do odczytu (zobacz punkt 2.2).
4. Zmień nazwy wybranych plików poleceniem `mv`. Przenieś katalog `abc/xyz` do katalogu bieżącego tworząc w ten sposób katalog `xyz` w katalogu domowym. Przenieś wybrany plik z katalogu `abc` do katalogu `xyz` zmieniając jednocześnie jego nazwę. Sprawdź działanie przełączników `-i` i `-f` polecenia `mv`.

### 2.2 Prawa dostępu

1. Zinterpretuj następujące prawa dostępu:

```
-rwxr-xr-x 1 daniel users 164 Feb 21 17:19 test2
-rw-rw---- 1 kamil students 24250 May 27 2002 dane.txt
-r-----r-- 1 marek users 28014 Feb 21 17:43 przyklad.jpg
-rwxrwxrwx 1 daniel students 4563 Mar 8 04:43 pomoc.html
-r-xr--r-- 1 witek users 4611 Mar 8 04:42 pasjans
```

- (a) Które pliki mogą być modyfikowane przez użytkownika `daniel` należącego do grup `users` i `students`?
  - (b) Które pliki mogą być czytane przez użytkownika `marek` będącego członkiem grupy `students`?
  - (c) Kto może wykonywać program `pasjans`?
2. Dla wybranych plików: dodaj prawo zapisu dla grupy, odejmij prawo zapisu dla właściciela, dodaj prawo do wykonywania dla wszystkich użytkowników, ustaw prawa użytkownika na `rw`, usuń wszystkie prawa dla grupy i pozostałych użytkowników, ustaw prawa dla wszystkich użytkowników na `rw`.
  3. Ustaw prawa dostępu do wybranych plików na takie, jak w przykładzie z punktu 1 korzystając z notacji numerycznej.
  4. Zweryfikuj swoje prawa dostępu do plików `/etc/passwd`, `/etc/shadow`, `/var/mail/xxx`, gdzie `xxx` to twoja nazwa użytkownika.
  5. Wykonaj polecenie `umask 077` i sprawdź w jaki sposób ma to wpływ na prawa dostępu do nowo tworzonych plików. Nowe pliki możesz utworzyć poleceniem `touch(1)`.
  6. Dodaj prawo do zapisu dla grupy dla całej struktury katalogów `abc` (działanie rekurencyjne). Komendę wykonaj kilkakrotnie z dodatkowym przełącznikiem `-c`.
  7. Zapoznaj się z dokumentacją do poleceń `chown(1)` i `chgrp(1)`. Zmień grupę, do której należy wybrany plik na jedną z tych, które są wymieniane poleceniem `id(1)`.
  8. Sprawdź prawa dostępu do plików `/usr/bin/passwd`, `/usr/bin/write` i katalogu `/tmp`.

## 2.3 Wyszukiwanie plików

1. Znajdź przy pomocy programu `locate(1)` wszystkie pliki, które zawierają w nazwie słowo `netscape`. Listę ogranicz najpierw do tych programów, które występują w podkatalogach katalogu `/usr`, a następnie tylko do tych plików, których nazwa brzmi dokładnie `netscape`. Znajdź 3 literowe programy znajdujące się w podkatalogach `bin` katalogu `/usr`.
2. Korzystając z programu `find`: wyszukaj w katalogu `/usr/share/doc` pliki z rozszerzeniem `TXT` (duże litery!), w katalogu `/usr` znajdź wszystkie podkatalogi o nazwie `bin`, w katalogu `/etc` znajdź wszystkie dowiązania symboliczne, znajdź programy z katalogu `/usr/bin` nie używane od ponad miesiąca, w katalogu domowym znajdź pliki zmodyfikowane dzisiejszego dnia, w podkatalogach `/usr/lib` znajdź pliki o rozmiarze przekraczającym 3 MB.
3. Skopiuj wszystkie pliki zwykłe o rozmiarze pomiędzy 1 a 100 bajtów z katalogu `/usr/bin` do katalogu `x` w katalogu bieżącym.

## 2.4 Dowiązania

1. Utwórz dowiązanie twarde do wybranego pliku. Zmień prawa dostępu do dowiązania i porównaj je z oryginalnym plikiem. Sprawdź nr i-węzła obu plików za pomocą polecenia `ls -li`. Utwórz dowiązanie twarde do katalogu. Usuń dowiązanie lub oryginalny plik obserwując zmianę licznika dowiązań (druga kolumna wyników `ls -li`).
2. Utwórz dowiązanie symboliczne do pliku. Sprawdź i zmień prawa dostępu do dowiązania. Usuń plik oryginalny i wyświetl programem `more` zawartość pliku-dowiązania. Utwórz dowiązanie symboliczne do katalogu. Utwórz dowiązanie symboliczne do nieistniejącego pliku. Sprawdź na co wskazują dowiązania symboliczne: `/dev/cdrom`, `/dev/mouse` i `/usr/X11`.

# 3

## Edytory

### 3.1 Edytor vi

1. Przećwicz nawigację po przykładowym dokumencie: komendy `j`, `k`, `h`, `l`, `0`, `$`, `G`. Wstawianie tekstu: komendy `i`, `a` i `R` zakończone wciśnięciem `Esc`. Przemieszczanie się do następnego/poprzedniego słowa: `w`, `e`, `b`.
2. Usuwanie tekstu: komendy `x`, `X` i `dd`. Wycofywanie ostatnich operacji: komenda `undo` `u` i `redo` `Ctrl-r`.
3. Wyszukiwanie tekstu komendami `/` i `?`. Posługując się wyrażeniami regularnymi wyszukaj w tekście puste linie, linie zaczynające się od znaku „#”, dowolne liczby, ciągi spacji dłuższe niż 1 znak.
4. Kopiowanie tekstu: komendy `p`, `P` i `yy`.
5. Parametryzowanie komend poprzez znaczniki, np.: `d0`, `dG`, `d$`. Komenda `c` i `y` w połączeniu ze znacznikami. Parametryzowanie numeryczne: `10G`, `5i`, `d5d`, `d2w`, `d3k`, `d10h`, `d/x`.
6. Operacje na plikach: zapisywanie `:w`, wstawianie zewnętrznego pliku `:r`. Opuszczenie edytora `:q`, `:q!`, z zapisem: `:x`, `ZZ`.
7. Konfiguracja edytora: komendy `:set number`, `:set autoindent`. Konfiguracja w pliku `~/ .exrc`.
8. Rozszerzenia edytora vim: zaznaczanie tekstu `V`, `v`, `Ctrl-v` i jedna z komend: `d`, `y`, `c`. Podświetlanie składni `:syntax on`.

### 3.2 Edytor emacs

1. Wczytaj dokument *Tutorial* komendą `Ctrl-h t`. Przemieszczaj się po tekście komendami `Ctrl-f`, `Ctrl-b`, `Ctrl-n`, `Ctrl-p`, `Ctrl-v`, `Alt-v`, `Ctrl-a`, `Ctrl-e`, `Alt-f`, `Alt-b`, `Alt-a`, `Alt-e`, `Alt->`, `Alt-<`. Usuwanie tekstu: `Ctrl-d`, `Ctrl-k`, `Alt-k`.



2. Kopiowanie tekstu: `Ctrl-Space`, przesunąć kursor, `Ctrl-w` i wstawić w innym miejscu: `Ctrl-y`. Wycofanie operacji: `Ctrl-x u` (lub `Ctrl-_`).
3. Wczytywanie pliku `Ctrl-x Ctrl-f`. Zapis pliku `Ctrl-x Ctrl-s`.
4. Wyszukiwanie tekstów: `Ctrl-s`, `Ctrl-r`.
5. Wyjście z edytora: komenda `Ctrl-x Ctrl-c`.

# 4

## Procesy

### 4.1 Lista procesów

1. Wyświetl listę własnych procesów komendą `ps`. Porównaj wyniki z wynikami poleceń: `ps -x` i `ps -ax`. Zbadaj działanie przełączników `-l` i `-u`. Zaloguj się do systemu kilkakrotnie poprzez wirtualne konsole lub otwierając nowe okno w środowisku graficznym. Sprawdź poleceniem `tty(1)` nazwę terminala, na którym pracujesz.
2. Znajdź proces macierzysty dla procesu `ps`. Odszukaj przodka wszystkich procesów (PID=1). Wyświetl hierarchię procesów poleceniem `ps tree(1)`. Obejrzyj listę procesów poleceniem `top(1)` sortując ją wg stopnia zajętości procesora i ilości zajętej pamięci. W środowisku graficznym KDE uruchom program `ksysguard` (dostępny po wciśnięciu `Ctrl-Esc`).
3. Sprawdź identyfikator procesu `init`. Za pomocą polecenia `pgrep(1)` wyświetl identyfikatory wszystkich swoich interpreterów poleceń.

### 4.2 Sygnały

1. Zapoznaj się z listą sygnałów na stronie pomocy systemowej `signal(7)`. Uruchom sesję edytora `vi` i wysyłaj komendą `kill` kolejne sygnały do tego procesu. Użyj sygnałów: HUP, INT, TERM, QUIT, KILL, STOP, CONT.
2. Zbadaj działanie poleceń `killall(1)` i `pkill(1)`.

### 4.3 Priorytety procesów

1. Obniż wybranemu procesowi priorytet poprzez zmianę wartości `nice` (komenda `renice`). Zmień priorytet interpretera poleceń i sprawdź priorytety procesów potomnych tego interpretera.
2. Uruchom nowe procesy ze zmienionym priorytetem (komenda `nice`).

## 4.4 Obsługa procesów w interpreterze poleceń

1. Wstrzymaj sesję edytora `vi` kombinacją `Ctrl-z`, uruchom nową sesję i wstrzymaj ją również. Wyświetl aktywne sesje komendą `jobs`. Wznów pracę do trybu pierwszoplanowego komendą `fg`.
2. Uruchom proces `sleep` w tle. Przełącz go do pracy w trybie pierwszoplanowym, a następnie znów przełącz go do pracy w tle komendą `bg`.

# 5

## Potoki

### 5.1 Filtr `cat`

1. Przecwicz przekierowania do i z plików z wykorzystaniem filtra `cat`.
2. Połącz zawartość plików `1.txt`, `2.txt` i `3.txt` w jeden plik `wynik.txt`.

### 5.2 Filtry `head`, `tail`

1. Korzystając z polecenia `cat` utwórz plik `tekst3`, który będzie składał się z zawartości pliku `tekst`, ciągu znaków podanego ze standardowego wejścia (klawiatury) i pliku `tekst2`.
2. Wyświetl po 5 pierwszych linii wszystkich plików w swoim katalogu domowym w taki sposób, aby nie były wyświetlane ich nazwy.
3. Wyświetl linie od 3 do 5 z pliku `/etc/passwd`.
4. Wyświetl linie od 5 do 3 licząc od końca z danych wyświetlanych przez polecenie `ls`.

### 5.3 Filtr `grep`

1. Wyświetl informacje o dowiązaniach symbolicznych z katalogu `/usr/bin`.
2. Wyświetl z pliku `tekst` linie nie zawierające żadnych liczb.
3. Wyświetl informacje o takich plikach zwykłych z katalogu bieżącego, które mają ustawione prawo zapisu dla grupy.
4. Wypisz nazwy plików zakończonych na `.txt` z katalogu domowego, które zawierają linie z komentarzem (linie zaczynające się od znaku „`#`”).
5. Przeszukaj swój cały katalog domowy w poszukiwaniu plików `txt` zawierających słowo `abc`.

## 5.4 Filtry `wc`, `tr`

1. Zlicz wszystkie pliki zwykle znajdujące się w katalogu `/etc` i jego podkatalogach (2 sposoby).
2. Napisać polecenie zliczające sumę znaków z pierwszych trzech linii pliku `/etc/passwd`.
3. Oblicz długość trzeciej linii pliku `/etc/passwd`.
4. Oblicz ilość pustych linii w pliku `/etc/modules.conf`.
5. Wyświetl zawartość `/etc/passwd` w jednej linii.
6. Za pomocą filtra `tr` wykonaj modyfikację pliku, polegającą na umieszczeniu każdego słowa w osobnej linii.
7. Wyświetl listę plików z aktualnego katalogu, zamieniając wszystkie małe litery na duże.

## 5.5 Filtry `cut`, `sort`, `uniq`

1. Wyświetl listę praw dostępu do plików w aktualnym katalogu, ich rozmiar i nazwę.
2. Wyświetl listę plików w aktualnym katalogu, posortowaną według rozmiaru pliku. Wyświetl same nazwy plików w tej samej kolejności.
3. Wyświetl zawartość pliku `/etc/passwd` posortowaną wg numerów UID w kolejności od największego do najmniejszego. Wyświetl zawartość pliku `/etc/passwd` posortowaną najpierw według numerów GID w kolejności od największego do najmniejszego, a następnie według UID.
4. Wyświetl liczby plików utworzonych przez poszczególnych użytkowników w podkatalogach katalogu `/tmp`.
5. Wypisz w kolejnych liniach 5 najczęściej powtarzających się słów z pliku o nazwie `tekst`.
6. Podaj nazwy trzech najmniejszych plików z bieżącego katalogu posortowane wg nazwy.
7. Wyświetl nazwy zalogowanych w systemie użytkowników, którzy mają uruchomiony program `vi`.
8. Podaj nazwy 5 użytkowników o największej liczbie uruchomionych procesów.
9. Wyświetl niepowtarzające się nazwy użytkowników będących właścicielami 10 procesów, które zajmują najwięcej pamięci w systemie.
10. Wyświetl statystykę używanych komend (bez argumentów) w postaci posortowanej listy: ilość komenda

# 6

## Interpreter poleceń

### 6.1 Zmienne środowiskowe

1. Zdefiniuj zmienną `X` i przypisz jej napis `linux`. Wyświetl zmienną komendą `echo`. Sprawdź obecność zmiennej w nowym interpreterze (procesie potomnym). Usuń zmienną. Ustaw zmienną `X` jako zmienną eksportowaną.
2. Wyświetl listę zmiennych eksportowanych. Zbadaj efekt zmiany wartości zmiennej `PS1`. Sprawdź działanie programów `vi` i `mc` po ustawieniu zmiennej `TERM` na jedną z wartości: `vt100`, `dumb`, `ansi`.

### 6.2 Aliasy i funkcje

### 6.3 Konfiguracja

# 7

## Programowanie w sh

### 7.1 Argumenty

1. W pliku zapisz zlecenie `echo Hello`, nadaj prawo do wykonywania pliku i uruchom skrypt. Dopisz komentarz do pliku. Wskaż ścieżkę do interpretera. Wyświetl wartości zmiennych pozycyjnych `$1`, `$2`, itd. Przećwicz działanie polecenia `shift`. Wywołaj skrypt z argumentami:

```
a b c
"a b" c
*
"*"
```

### 7.2 Instrukcja warunkowa

1. Napisz skrypt sprawdzający czy istnieje podany jako parametr plik i wypisujący odpowiedni komunikat na ekranie.
2. Napisz skrypt przyjmujący jako parametry wejściowe następujące wartości: nazwę pliku, ścieżkę do katalogu i nową nazwę pliku. Skrypt ma za zadanie powielić wskazany plik w podanym katalogu i wszystkich jego podkatalogach (1 poziom włącznie) informując o tym użytkownika. W przypadku błędnie podanych parametrów wejściowych należy poinformować użytkownika.

### 7.3 Pętle `for` i `while`

1. Napisz skrypt wyświetlający w kolejnych liniach argumenty przekazane do skryptu.
2. Napisz skrypt sortujący zawartości wszystkich przekazanych argumentami plików.
3. Napisz skrypt kopiujący plik wskazany pierwszym argumentem do wszystkich katalogów wskazanych kolejnymi argumentami. Przykładowe wywołanie skryptu:

```
# kopiuj.sh dane.txt d1 d2 d3
```

4. Napisz skrypt łączący trójkami pliki, których nazwy zostały podane jako parametry wejściowe, tj. zawartość pliku drugiego i trzeciego jest dołączana do zawartości pliku pierwszego, zawartości plików piątego i szóstego są dołączane do pliku czwartego, itd.
5. Napisz skrypt, który będzie wykonywał się w każdym z podkatalogów wskazanego katalogu i wypisywał nazwy plików zwykłych z tych katalogów.

## 7.4 Wczytywanie wartości

1. Napisz skrypt wyświetlający w nieskończonej pętli menu umożliwiające realizację wywołań komend `ls`, `ps` i `who`. Do prezentacji menu zdefiniuj funkcję.
2. Wyświetl informacje o procesach należących do użytkowników o nazwach wymienionych w kolejnych liniach wskazanego argumentem pliku. Informacje wyświetl w dwóch wyrównanych kolumnach: nazwa użytkownika i PID procesu.
3. Przetestuj działanie programu `dialog(1)` zapoznając się z pomocą dostępną po wyspecyfikowania przełącznika `--help`:

```
# dialog --help
```

## 7.5 Zadania zaawansowane

1. Napisz skrypt zmieniający rozszerzenia wszystkich plików w katalogu bieżących z wartości wskazanej pierwszym argumentem na wartość wskazaną drugim argumentem. Skorzystaj z polecenia `basename(1)`. Przykładowe wywołanie:

```
# zmien.sh txt doc
a.txt => a.doc
dane.txt => dane.doc
```

2. Napisz skrypt wyświetlający informacje o plikach zwykłych w następującym formacie:

właściciel atrybuty rozmiar nazwa

przy czym pole `właściciel` może przyjmować następujące wartości: `moj`, jeśli plik należy do użytkownika, `administrator` jeśli plik należy do użytkownika o nazwie `root` i `inni` w pozostałych przypadkach.

3. Napisz skrypt wyświetlający w odwrotnej kolejności argumenty jego wywołania, np.

```
# skrypt a b c d
d c b a
```

4. Napisz skrypt sortujący zawartości wszystkich plików o rozszerzeniach wskazanych pierwszym argumentem i znajdujących się w podkatalogach katalogów wskazanych pozostałymi argumentami.



5. Napisz skrypt zapisujący do pliku o nazwie `plik.usr` pełną informację o użytkownikach którzy mają odblokowane przyjmowanie komunikatów na terminal.
6. Napisz skrypt sprawdzający czy użytkownicy o identyfikatorach podanych jako parametry wejściowe są zalogowani w systemie więcej niż raz. Dla każdego takiego użytkownika należy wyświetlić jego identyfikator i listę terminali na których pracuje. Informacje o kolejnych użytkownikach powinny być oddzielone ciągiem gwiazdek.