

Wydział Zarządzania i Modelowania Komputerowego

Specjalność: Informatyka Stosowana

Rok III Semestr V

System operacyjny UNIX Ćwiczenie 7. Wyposażenie powłoki. Procesy.

1. Zmienne powłoki

- Zaobserwuj skutki zmian wartości niektórych zmiennych powłoki. Wykonaj polecenia:

```
set | more                # wnotuj wartości zmiennych HOME i PWD
zapas=$PATH              # zdefiniowano nową zmienną
set | more                # odzyskaj zmienną w spisie
echo $zapas               # wyświetlenie wartości zmiennej
PATH=""
cd ..
pwd
cd
set
ls                        # zmienna PATH pusta; działają tylko polecenia
                           # wewnętrzne tj. takie jak: pwd, set, cd

PATH=$zapas
unset zapas
echo PATH=$PATH  zapas=$zapas  # zmienna PATH odtworzona; zmienna zapas
                               # usunięta

mkdir k1
HOME=$HOME/k1
echo $HOME               # zmienna HOME ma nową zawartość
cd
pwd                       # działanie cd wg nowej wartości HOME
cd ..
HOME=$PWD                # odtworzenie starej wartości HOME
cd k1
cd
pwd                       # działanie cd jak poprzednio
PWD=/
echo $PWD                # nowa wartość zmiennej PWD
pwd
cd k1
pwd                       # zmiana PWD nie wpłynęła na działanie polecenia cd
echo $PWD                # system odtworzył wartość zmiennej PWD
cd
```

Zadanie 1

Sprawdź działanie polecenia **sleep**. Parametr polecenia określa długość okresu „uśpiania” powłoki w sekundach. Określ, czy polecenie to należy do grupy poleceń wewnętrznych – uruchom polecenie przy pustej wartości zmiennej PATH.

- Prześledź działanie polecenia *export* wykonując poniższe instrukcje:

```
A=Jeden
B=Dwa
HOME=$HOME/k1
set | more                # sprawdź wartości zmiennych A, B i HOME
export B
ps                        # tylko 1 proces powłoki
```

```

bash
ps                               # 2 procesy powłokowe
set | more                       # zmiennej A brak; B i HOME jak w powłoce głównej
A=Bolek
B=Lolek
set | more
exit
ps                               # 1 proces; exit spowodowało zakończenie procesu powłoki potomnej
set | more                       # „stare” wartości zmiennych
unset A B
HOME=$PWD

```

Zadanie 2

Napisz skrypt wyświetlający wartość zmiennej ALFA. Jaki jest rezultat wykonania skryptu w przypadku jeśli przed wywołaniem skryptu:

- zmiennej ALFA nie nadano żadnej wartości,
- zmiennej ALFA nadano wartość np. „Przykładowy napis”,
- zmiennej nadano wartość i określono ją jako „exportową”,
- zniesiono definicję eksportowania zmiennej (poleceniem **export** z opcją **-n**), a skrypt jest wywoływany z wykorzystaniem polecenia **source** (krótszy zapis polecenia to **.**).

2. Polecenie alias

- Wykonaj polecenia ilustrujące działanie polecenia alias:

```

alias                             # wyświetlenie listy aliasów
rm -r ?                          # skasowanie plików o nazwach jednoliterowych
touch a b
rm ?                              # nie kasuj (odpowiadaj n - NIE)
unalias rm
alias                             # brak aliasu dla rm
rm ?
ls -l                             # pliki a i b skasowane bez zadawania pytań

```

- Odtwórz alias dla polecenia rm, tak by polecenie to wykonywało się z opcją **-i**.

Zadanie 3

Zdefiniuj alias dla polecenia **cd** w taki sposób by po wykonaniu polecenia dodatkowo wyświetlać pełną ścieżkę katalogu docelowego.

3. Skrypty startowe

- Przeglądnij zawartość konfiguracyjnych skryptów startowych znajdujących się w katalogu osobistym każdego użytkownika: **.bash_profile** i **.bashrc**

Zadanie 4

Zmień zawartość pliku **.bashrc** dodając na końcu linię wyświetlającą tekst „Nowa powłoka” oraz linię kreującą nowy alias:

```
alias kat='echo Katalog `pwd` ; ls -l'
```

Sprawdź, że alias nie działa w powłoce bieżącej natomiast jest dostępny po uruchomieniu powłoki potomnej poleceniem: **bash**. Zakończ proces powłoki potomnej poleceniem **exit**.
Usuń wstawione linie z pliku .bashrc

4. Procesy i zadania

- Prześledź działanie poleceń: **ps**, **jobs**, **^Z**, **fg** wykonując polecenia:

```
man ps
ps
ps f
ps ax
ps r
jobs                # brak zadań w tle
cat > a             # wprowadź linię; zakończ ^Z – wstrzymanie procesu
cat a              # plik powstał
ps f               # proces cat został wstrzymany
jobs               # 1 zadanie wstrzymane
man ps >info.ps
^Z                 # zatrzymaj przed zakończeniem
jobs               # 2 zadanie wstrzymane
ps f               # wiele procesów zależnych od siebie
fg %2              # przeniesienie zadania 2 na pierwszy plan
jobs
fg                 # przeniesienie bieżącego zadania na pierwszy plan;
                  # dopisz linię i zakończ ^D
jobs               # brak zadań w tle
cat a              # plik a w całości
```

- Prześledź działanie poleceń: **bg**, **kill**. Zapisz w skrypcie o nazwie *wtle.bash* następujące polecenia:

```
echo Plik zawiera datę i czas wpisywane co 20 sekund > $1
while :
do
  echo Zapisywanie do pliku $1
  date >> $1
  sleep 20
done
```

Nadaj skryptowi uprawnienia do wykonania, a następnie wykonaj instrukcje:

```
ps f                # tylko proces powłoki i polecenia ps
wtle.bash plik1
^Z
cat plik1
ps f                # 2 nowe procesy o statusie T - wstrzymane
jobs                # 1 zadanie - wstrzymane
wtle.bash plik2 &  # uruchomienie zadania w tle (na drugim planie)
cat plik2
ps f                # 2 nowe procesy o statusie S - uśpione
jobs                # nowe zadanie nr 2 - pracujące
bg %1               # przeniesienie zadania 2 do pracy w tle
ps f
jobs                # oba zadania pracujące w tle
kill %2             # zakończenie zadania 2
ps f                # wynotuj nr procesu powłoki działającego skryptu
jobs -l             # też wyświetla nr procesu
kill <nr>           # w miejsce <nr> wpisz wynotowany nr procesu
ps
jobs                # brak zadań w tle
```

Zadanie 5

Napisz skrypt w którym uruchamiane jest w tle polecenie **man ls >/dev/null &**, a następnie 5-krotnie w odstępach 0.1 sek polecenie **ps f**. Dla odpowiedniego opóźnienia wykonania poleceń **ps** użyj polecenia **sleep**. Wyniki wyświetlane poleceniami **ps** przesyłaj do pliku o nazwie określonej parametrem skryptu.

Uruchom skrypt, a następnie przejrzyj zawartość pliku wynikowego.

- Prześledź mechanizm przesyłania sygnałów do procesów. Utwórz skrypt o nazwie *sygnaly.bash* i zawartości

```
echo Uruchomiono skrypt ktory odbiera sygnaly 15 (TERM) 40 i 41
echo Sygnaly 15 (TERM) i 40 koncza skrypt
echo Sygnal 41 powoduje wykonanie polecenia ls
trap exit 40
trap ls 41
while :
do
ls >/dev/null
done
```

Wykonaj polecenia:

```
sygnaly.bash &
ps                               # wynotuj nr procesu powłoki skryptowej
jobs
kill -TERM <nr>                  # wpisz nr procesu
ps
sygnaly.bash &                  # wynotuj nr procesu powłoki skryptowej
ps
jobs
kill -41 <nr>                    # wpisz nr procesu
kill -40 <nr>
ps
jobs
```

- Prześledź działanie mechanizmu cyklicznego wykonania zleceń. Utwórz plik o nazwie *cr* zawierający linię zlecenia okresowego w postaci:

```
* * * * * echo Data i czas: `date` >> /home/ksd_s1/plik
```

Uwaga: Pierwsze 5 gwiazdek to wzorzec czasu wykonania (co minutę). W zapisie następującego dalej polecenia wykonywanego cyklicznie należy podać kompletną ścieżkę dostępu. W przykładzie powyżej wyznacza ona plik w katalogu osobistym użytkownika *ksd_s1*.

Wykonaj polecenia:

```
touch plik
crontab -l                       # brak zleceń okresowych
crontab cr                       # skopiowanie cr do pliku crontab
crontab -l                       # jest 1 zlecenie
cat plik                         # plik jest tym dłuższy im więcej mija czasu
crontab -r                       # usunięcie zlecenia
```