

Ćwiczenie 3

Cel ćwiczenia: zapoznanie się z obsługą edytora vi, Poznanie podstaw pisania skryptów w powłoce bash.

Wykorzystane polecenia

vi, vim - uruchamia edytor tekstu

at - uruchamia polecenia o podanym czasie

date - wyświetla bieżącą datę i godzinę

sleep - pauza na określony czas podany w sekundach

Edytor vi.

Vi jest najpopularniejszym edytorem tekstu w systemach uniksowych. Można go znaleźć praktycznie w każdym systemie. W systemie Linux dostępny jest edytor vim (vi improved) Edytor vi uruchamiamy w bardzo prosty sposób podając jego nazwę wraz z opcjonalną nazwą pliku, który chcemy utworzyć lub otworzyć.

vi pliktekstowy

Vi pracuje w trybie komend i w trybie wpisywania. Niektóre komendy wywoływane są wciśnięciem odpowiednich klawiszy lub poprzez wpisanie ich w linii poleceń. Aby dostać się do linii poleceń należy użyć znaku dwukropka. Po dokumencie można poruszać się za pomocą strzałek jak również klawiszy:

k - góra

j - dół

h - lewo

l - prawo

Podstawowe komendy vi

i - rozpoczęcie wpisywania w miejscu w którym znajduje się kursor

a - rozpoczęcie wpisywania od następnego znaku za kursorem

A - rozpoczęcie wpisywania na końcu wiersza

c - usunięcie wiersza i rozpoczęcie wpisywania

o - wstawienie wiersza pod tym w którym znajduje się kursor i rozpoczęcie wpisywania

x - usunięcie znaku w miejscu kursora

dd - wycięcie wiersza (przeniesienie do bufora)

dw - wycięcie od miejsca w którym znajduje się kursor do końca słowa wraz ze spacją

de - wycięcie od miejsca w którym znajduje się kursor do końca słowa bez spacji

d\$ - wycięcie od kursora do końca wiersza

p - wklejenie zawartości bufora

Jeśli komenda wymaga pewnej ilości powtórzeń to można podać przed komendą odpowiednią liczbę. Na przykład 15x usunie piętnaście znaków, 4dd wytnie 4 wiersze poczynając od kursora w dół.

:q - wyjście (działa tylko w przypadku braku zmian w pliku)
:q! - wyjście z pominięciem zmian
:w - zapis
:wq - zapis i wyjście
:x -zapis i wyjście
ZZ - zapis i wyjście

W edytorze vi możliwe jest zapisanie dowolnego fragmentu tekstu z dokumentu do innego pliku:

```
:n,m w nazwa_pliku
```

gdzie n i m to numer pierwszego i ostatniego wiersza.
W dowolne miejsce dokumentu można wstawić dowolny plik:

```
:r nazwa_pliku
```

Pomoc dla programu vi dostępna jest po wpisaniu *:help*

Skrypty.

Skrypty w systemie Linux można porównać do plików wsadowych z systemu DOS (.bat). Jednak skrypty Linuksowe są o wiele bardziej rozbudowane i stwarzają więcej możliwości. Są to zwykle pliki tekstowe zawierające m. in. polecenia powłoki. W skryptach mogą znajdować się także instrukcje warunkowe, pętle, działania arytmetyczne. Skrypty można uruchamiać na trzy różne sposoby:

1. Poprzez określenie powłoki w której ma on zostać uruchomiony na przykład *sh skrypt parametry*
2. Poprzez nadanie praw do wykonywania.
3. Używając znaku kropka do uruchomienia skryptu na przykład: *. skrypt parametry*

W skryptach można używać zmiennych. Do oznaczania parametrów pozycyjnych (podawanych za nazwą skryptu przy uruchamianiu) używamy numerów poprzedzonych znakiem \$. Przy czym:

\$0 - nazwa skryptu

\$numer - kolejny numer parametru.

Aby odwołać się do parametru o numerze większym od 9 należy użyć nawiasów klamrowych na przykład:

```
${10}
```

Aby przypisać zmiennej wartość równą działaniu arytmetycznemu na liczbach należy użyć polecenia *let*. Na przykład wyświetlenie sumy dwóch liczb:

```
Z1=10
```

```
Z2=20
```

```
let ZS=$Z1+$Z2
```

```
echo $ZS
```

Warunki.

Do sprawdzania warunków służy instrukcja *if*. Jej format wygląda następująco

```
if [ test ]
then
    polecenia
else
    polecenia
elif [ test ]
then
    polecenia
fi
```

[-d plik]	- plik istnieje i jest katalogiem
[-f plik]	- plik istnieje i plikiem zwykłym
[-r plik]	- plik istnieje i mamy do niego prawo czytania
[-w plik]	- plik istnieje i mamy do niego prawo pisania
[-x plik]	- plik istnieje i mamy do niego prawo wykonywania
[-s plik]	- plik istnieje i mam rozmiar większy niż zero bajtów
[-z s1]	- długość łańcucha s1 jest zerowa
[-n s1]	- długość łańcucha s1 nie jest zerowa
[s1 = s2]	- łańcuchy s1 i s2 są identyczne
[s1 != s2]	- łańcuchy s1 i s2 nie są identyczne
[s1]	- łańcuch s1 nie jest pusty

Należy zwrócić uwagę na znak spacji pomiędzy nawiasami kwadratowymi a testem warunku. Jest to często popełniany błąd przy pisaniu skryptów.

Instrukcja *elif* przeprowadza test gdy pierwszy warunek nie zostanie spełniony.

Możemy tu używać operatorów logicznych takich jak AND (-a) OR (-o) NOT (!). Na przykład:

```
if [ -d plik -a -f plik2 ]; then
    mv plik plik3
    mv plik2 plik
else
    touch plik
fi
```

Innym rodzajem instrukcji warunkowej jest instrukcja *case*. Jej format to:

```
case $1 in
    wartosc1) polecenia ;;
    wartosc2) polecenia ;;
    wartosc3) polecenia ;;
esac
```

Ćwiczenia.

1. Przeczytaj pomoc man na temat wszystkich poleceń użytych w ćwiczeniu.
2. Przeczytaj pomoc man na temat powłoki *bash*.
3. Wyprowadź do pliku *lista* zawartość katalogu */usr/local*, a następnie wykorzystując edytor *vi* zapisz 4 ostatnie wiersze do pliku *koniec*.
4. Wyprowadź do pliku *lista_etc* zawartość katalogu */etc* Wstaw ten plik na początek pliku *koniec* z poprzedniego ćwiczenia używając *vi*.
5. Napisz skrypt wyświetlający nazwę skryptu oraz cztery jego parametry.
6. W katalogu domowym stwórz plik o nazwie *data*. Stwórz z niego skrypt wyświetlający datę w formacie RRRR-MM-DD.
7. Stwórz skrypt nadający prawo do zapisu dla grupy, plikowi podanemu jako parametr (wywołanie: *skrypt nazwapliku*). Jeśli nazwa pliku nie zostanie podana wyświetl komunikat o błędzie.
8. Załóżmy, że w systemie nie ma polecenia *mv*. Napisz skrypt, który umożliwi zastąpienie tego polecenia.
9. Napisz skrypt obliczający sumę, iloczyn, iloraz 4 liczb podanych jako parametry.
10. Stwórz skrypt wyświetlający grupami pliki do których mamy prawo do odczytu, zapisu i wykonania.
11. Napisz skrypt, który po 1 minucie od uruchomienia wyświetli napis „Minela jedna minuta” (polecenie *at*)
12. Napisz skrypt wyświetlający trzy komunikaty w odstępach pięciosekundowych.
13. Napisz skrypt dodający do podanej nazwy pliku rozszerzenie *.old*
14. Znajdź błędy w skrypcie:

```
if [-d plik ]; then
    mv plik plik2
elif [ -f plik3 ] then
    mv plik3 plik4
fi
```

Opisz działanie skryptu.

15. Stwórz skrypt, który wyszuka nazwę procesu podaną jako parametr wśród wszystkich procesów obecnych w systemie i wyświetli informacje dotyczące tego procesu.
16. Napisz skrypt, który w zależności od podanego parametru wyświetli następujące komunikaty:
 - *pomoc*: „Nie ma dostępnej pomocy”
 - *usun*: „Nie ma nic do usunięcia”
 - *kopiuj*: „Nie ma plików do kopiowania”Jeśli podany parametr nie będzie pasował do wymienionych należy wyświetlić jego nazwę.
17. W pliku *.bash_logout* umieść polecenie usuwające wszystkie pliki z katalogu domowego. Wyloguj się i zaloguj ponownie. Jaki efekt?
18. Napisz skrypt przeszukujący plik *.bash_history* i wyświetlający wszystkie wystąpienia polecenia *cat*

Pytania:

1. Jak zapisać w programie *vi* dowolny fragment tekstu do pliku?
2. Jak wyświetlić informacje o pozycji kursora w dokumencie?
3. Czy skrypt może być wywoływany z innego skryptu?
4. Jak deklarować zmienne w skryptach?
5. Kiedy wykonuje się skrypt *.bash_logout*, a kiedy *.bash_profile*?