

Ćwiczenie 4

Cel ćwiczenia: poznanie mechanizmów działania pętli w skryptach, zapoznanie się z obsługą edytora joe, poznanie poleceń do wykonywania operacji na tekście.

Wykorzystane polecenia

cut – wycina z tekstu żądane fragmenty informacji

wc – program wypisujący informacje o pliku: liczbę linii, słów oraz bajtów.

who – wyświetla aktualnie zalogowanych użytkowników w systemie

seq – wypisuje liczby od 1 do zadanej.

Edytor JOE

Joe jest kolejnym popularnym edytorem dostępnym w wielu systemach Linux. Uruchamiamy go pisząc *joe* w linii poleceń. Dla edytora tego dostępna jest pomoc *man* oraz pomoc w programie, którą można przejrzeć wciskając *CTRL+k h*. Program umożliwia operacje na blokach tekstu, wyrazach liniach, wstawianie plików itp. Podstawowe klawisze używane w programie:

Poruszanie się po tekście:

CTRL+Z – poprzedni wyraz

CTRL+X – następny wyraz

CTRL+A – początek linii

CTRL+E – koniec linii

CTRL+U – poprzedni ekran

CTRL+V – następny ekran

CTRL+KU – początek tekstu

CTRL+KV – koniec tekstu

Operacje na blokach tekstu:

CTRL+KB – początek bloku

CTRL+KK – koniec bloku

CTRL+KM – przeniesienie bloku

CTRL+KC – skopiowanie bloku

CTRL+KW – zapisanie bloku do pliku

CTRL+KY – usunięcie bloku

Usuwanie tekstu:

CTRL+D – usunięcie znaku

CTRL+Y – usunięcie linii

CTRL+W – usunięcie słowa na prawo od kursora

CTRL+O – usunięcie słowa na lewo od kursora

CTRL+J – usunięcie tekstu do końca linii na prawo od kursora

Operacje na plikach:

CTRL+KD – zapisanie pliku

CTRL+KR – wstawienie pliku

CTRL+KE – edycja pliku

Wyjście z programu:

CTRL+KX – wyjście z zapisem

CTRL+C – wyjście bez zapisu

Funkcje.

Podobnie jak w programach w skryptach można definiować funkcje, które mogą zawierać często powtarzające się operacje. Format definiowania funkcji wygląda następująco:

```
function nazwa
{
    Polecenie1
    Polecenie2
}
```

Do funkcji można przekazywać parametry podobnie jak przy wywołaniu skryptu. Wewnątrz funkcji widoczne są one jako \$1, \$2 itd.

Wczytywanie danych.

Skrypty umożliwiają interakcję z użytkownikiem. Do wczytywania danych wprowadzanych z klawiatury służy polecenie *read*.

read - wczytuje linie ze standardowego wejścia i wysyła je do zmiennej \$REPLY.

read ZMIENNA - wczytuje linie ze standardowego wejścia i wysyła je do zmiennej \$ZMIENNA

read Z1 Z2 Z2- wczytuje linie ze standardowego wejścia i wysyła je do kolejnych zmiennych (jedna linia w jednej zmiennej)

read < plik – wczytuje dane z pliku.

Pętle.

Pętle mają podobną składnię jak w popularnych językach programowania.

Format pętli while:

```
while test
do
    polecenie1
    polecenie2
done
```

Format pętli for:

```
for i in test; do
    instrukcja1
    instrukcja2
done
```

Zmienna i przyjmuje kolejne wartości wynikające z testu. Na przykład:

```
for i in /usr/local/*
do
    echo $i
done
```

spowoduje wyświetlenie zawartości katalogu */usr/local*. Aby wykonać jakieś polecenie określoną ilość razy należy użyć polecenia *seq* w pozycji test. Aby umieścić w pozycji test program wykonywalny, którego dane wyjściowe będą przyjmowane przez zmienną *\$i* należy użyć apostrofów przy znaku „~”. Na przykład:

```
for i in `cat /etc/issue`  
do  
    echo $i  
done
```

Do tworzenia menu przydatna może być pętla *select*, która działa dopóki nie zostanie wywołane polecenie *break* lub *return*. Tworzy ona ponumerowaną listę. Jej format to:

```
select m in pozycja1 pozycja2  
do  
    polecenie 1  
    polecenie 2  
done
```

W pętli *select* często korzysta się z instrukcji *case*. Na przykład

```
select i in Kopiuj Przenies Koniec  
do  
    case $i in  
        "1") cp plik plik1 ;;  
        "2") mv plik plik2 ;;  
        "3") exit ;;  
        *) echo Nic nie wybrales ;;  
    esac  
done
```

Ćwiczenia.

1. Przeczytaj pomoc *man* na temat wszystkich wymienionych poleceń. W ćwiczeniach korzystaj z edytora *joe*
2. Z pliku */etc/services* wydobądź blok tekstu zaczynający się na *tcpmux*, a kończący się na *telnet*. Zapisz go do pliku o nazwie *uslugi* w katalogu domowym. Z pliku */etc/inetd.conf* zapisz pierwsze 10 linii do pliku *inet* w katalogu domowym. Stwórz nowy plik o nazwie *info*. Wstaw do niego pliki *uslugi* oraz *inet*. W pliku *info* opisz który fragment tekstu pochodzi z którego pliku. Do wszystkich operacji użyj edytora *joe*.
3. Napisz skrypt, który dla wszystkich plików w katalogu domowym ustawi prawo do odczytu dla wszystkich.
4. Napisz skrypt wypisujący tylko ilość słów w pliku tekstowym podanym jako parametr.
5. Stwórz skrypt wykonujący działanie na dwóch liczbach. Po podaniu liczb działanie powinno być wybrane z listy.
6. Napisz skrypt, który umożliwi dodawanie rozszerzenia do pliku. Skrypt powinien najpierw spytać o nazwę pliku, a następnie wyświetlić listę rozszerzeń (np.: *.old*, *_bak*, *.-001*) do wyboru.
7. Napisz skrypt wyświetlający pliki w podanym katalogu. Katalog ma być podawany po uruchomieniu skryptu.
8. Napisz skrypt wyświetlający same nazwy użytkowników aktualnie zalogowanych w systemie.
9. Napisz skrypt wyświetlający rozmiar pliku, podanego jako parametr, w bajtach.

Pytania.

1. Jak uzyskać pomoc w programie *joe*?
2. Jak wyjść z programu *joe* i zapisać zmiany?
3. Jak skopiować blok tekstu za pomocą edytora *joe*?
4. Co jest na ostatniej pozycji w wyniku działania polecenia *who*?
5. W jakiej kolejności polecenie *wc* wyświetla informacje?

Do sprawozdania.

Opisać przebieg ćwiczeń. Odpowiedzieć na pytania.