

### Chap III

Wyobraźmy sobie sytuację, gdzie mamy za zadanie napisanie prostego menu z kilkoma opcjami do wyboru, rozwiązanie problemu wyboru opcji na warunku if wydaje się bezsensowne ze względu na zmuszenie do zapisania dużej ilości if ów co powoduje brak czytelności kodu. Rozwiązanie tego problemu za pomocą instrukcji case będzie o wiele lepszym i czytelniejszym rozwiązaniem.

Składnia case'a wygląda następująco:

```
case zmienna in
  "wzorzec1") polecenie1 ;;
  "wzorzec2") polecenie2 ;;
  "wzorzec3") polecenie3 ;;
  *) polecenie_domyślne
esac
```

Jest ona bardzo prosta do zrozumienia. Pierwsza linijka zaczynająca instrukcję zawiera wartość zdefiniowaną przez nas w zmiennej, następnie linijki są wzorcami sprawdzającymi poprawność z wartością, jeśli któraś okaże się zgodna, wtedy następuje wykonanie poleceń przypisanych do tego wzorca i zakończenie instrukcji. Jeśli żaden ze wzorców nie będzie pasować, następuje wtedy wykonanie polecenia domyślnego, które oznaczone jest przez \* i zakończenie działania instrukcji. Aby składnia była prawidłowa, należy na końcu listy wzorców zapisać komendę esac.

Prezentując na przykładzie:

```
#!/bin/bash
echo "Menu:"
echo "1. Aktualna data"
echo "2. Zalogowani użytkownicy"
echo "3. Data włączenia systemu"
read d
case "$d" in
  "1") echo "Data: "; date ;;
  "2") echo "Użytkownicy: "; users ;;
  "3") who -b ;;
  *) echo "Błędny wybór"
esac
```

Proste menu, prosto i czytelnie zapisane. Wpisując liczbę 1,2 lub 3 trafimy na któryś ze wzorców i zostaną wykonane przypisane mu polecenia (jak widać we wzorcu 1 i 2, można zapisać więcej niż jedno polecenie, wystarczy oddzielić je średnikiem). Wpisanie czegokolwiek innego spowoduje wyświetlenie „Błędny wybór”.

Założmy, że posiadasz zbliżoną komendę w której zmienia się tylko jedna zmienna, po co więc niepotrzebnie rozpisywać się próbując zadeklarować wszystkie możliwe przypadki użycia, kiedy można użyć pętli select.

Składnia select jest bardzo prosta:

```
select zmienna in lista
do
  polecenie
done
```

Prezentując na przykładzie:

```
#!/bin/bash
echo "Wybierz katalog którego zawartość chcesz obejrzeć:"
select y in "/home" "/media" "/mnt"
do
  ls -la $y
  break
done
```

Uruchamiając taki skrypt, wyświetlone zostanie pytanie oraz 3 opcje do wyboru:

Wybierz katalog którego zawartość chcesz obejrzeć:

```
1) /home
2) /media
3) /mnt
#?
```

Aby dokonać wyboru wpisujemy odpowiednią cyfrę, zostanie wykonane polecenie (w tym wypadku `ls -la`) dla tej zmiennej. Za poleceniem `ls` znajduje się `break`, dlaczego? Pętla `select` jest pętlą wieczną, wyjście z niej możliwe jest tylko poprzez przerwanie jej.

Powróćmy jeszcze do problemu przedstawionego przy instrukcji `case`, znając `select` można uprościć cały wcześniejszy skrypt:

```
#!/bin/bash
echo "Menu:"
select y in "Aktualna data" "Zalogowani użytkownicy" "Data włączenia systemu" "Wyjście"
do
  case "$y" in
    "Aktualna data") echo "Data: "; date ;;
    "Zalogowani użytkownicy") echo "Użytkownicy: "; users ;;
    "Data włączenia systemu") who -b ;;
    "Wyjście") exit ;;
    *) echo "Błędny wybór"
  esac
break
done
```