

# Systemy operacyjne

## Laboratorium 6

### Tworzenie i uruchamianie skryptów powłoki *bash*

#### **Powtórzenie:**

cd /ścieżka do nowego katalogu – zmiana katalogu

cd .. – zmiana katalogu (piętro wyżej)

cd – przeniesienie do katalogu domowego

ls – zwraca zawartość katalogu roboczego

ls -l – zwraca zawartość katalogu z dodatkowymi informacjami

ls -a – zwraca zawartość katalogu razem z plikami ukrytymi

man – manual

date – wyświetla datę i godzinę

cp – kopiuje pliki i katalogi

mv – przenosi pliki i katalogi lub zmienia ich nazwy

mkdir – tworzy katalogi

rm – usuwa pliki i katalogi

chmod – zmienia ustawienia dostępu do pliku/katalogu

chown – zmienia właściciela pliku

chgrp – zmienia grupę do jakiej należy plik

umask – nadawanie domyślnych ustawień praw dostępu dla nowych plików i katalogów

df – zwraca informacje na temat zajętego i dostępnego miejsca

uname – zwraca podstawowe informacje o systemie

hostname – zwraca nazwę hosta oraz

free – zwraca informacje na temat zajętego i wolnego miejsca w pamięci

lslogins – wypisuje informacje na temat wszystkich użytkowników systemu

grep – szukanie plików według ich zawartości

find – szukanie plików według atrybutów

## 1. Wstęp

Skrypt shellowy można postrzegać jako zaawansowaną formę listy określonych komend, które mają być wykonane przez system operacyjny. Istnieje powszechnie akceptowana praktyka, według której każdy skrypt uruchamiany w środowisku BASH powinien rozpoczynać się od specyficznej linijki: `#!/bin/bash`. Ta linia ma kluczowe znaczenie, ponieważ za jej pomocą określamy shell, który ma odpowiadać za przetworzenie i wykonanie zawartych w skrypcie instrukcji. Chociaż pominięcie tej linii nie uniemożliwi działania skryptu, jej obecność jest zalecana. Różne shelle mogą posiadać nieco odmienne składnie, co oznacza, że skrypt bez tej linii może zachowywać się nieprzewidywalnie – czasami działa poprawnie, a innym razem nie – w zależności od tego, w jakiej powłoce zostanie uruchomiony.

W procesie nauki programowania, pierwszym zadaniem dla osób pragnących zgłębić tajniki danego języka programowania jest napisanie programu, który wyświetla krótki tekst na ekranie monitora. Aby trzymać się tych „powszechnie przyjętych norm”, również w Bashu pierwszym skryptem, który zostanie przedstawiony w tym kursie, będzie skrypt wyświetlający na ekranie monitora krótki tekst. Skrypt ten jest niezwykle prosty i składa się z dwóch linijek kodu:

```
#!/bin/bash
```

```
echo "Witaj świecie"
```

Uruchomienie tego skryptu spowoduje wyświetlenie na ekranie monitora komunikatu o treści: „Witaj świecie”, a następnie przejście do następnej linii. Aby pozostać w tej samej linii, należałoby przy poleceniu `echo` dodać parametr `-n`, co wyglądałoby tak: `echo -n "Witaj świecie"`.

Jak można się łatwo domyślić, polecenie `echo` odpowiada za wyświetlenie tekstu na ekranie monitora. Bardzo częstą praktyką wśród programistów jest komentowanie poszczególnych linii powstającego kodu. Także w Bashu jest to możliwe, a dokonuje się tego przy użyciu znaku `#`, co w rezultacie wygląda następująco:

```
#!/bin/bash
```

```
echo "Witaj świecie" # komentarz opisujący kod
```

Dla wielu osób, które miały już wcześniej styczność z programowaniem od razu rzuca się w oczy, że ta instrukcja nie jest zakończona znakiem średnika, a samo źródło skryptu nie znajduje się pomiędzy słowami kluczowymi np. `begin` i `end`. Są to jedne z wielu różnic, które występują między „zwykłym programowaniem” a pisanem skryptów w powłoce Bash.

## **ZADANIA** (zapisz odpowiedzi oraz użyte komendy w notatniku)

1. Napisać skrypt `kod.sh` który będzie pytał użytkownika o kod (który sam zdefiniujesz). Jeśli będzie poprawny to wyświetli zawartość pliku `dane.txt`. W przypadku złego hasła będzie trzy razy pytał ponowne wpisanie poprawnego hasła. Za trzecim niepoprawnym skrypt się wyłączy.
2. Napisać skrypt `dorosly.sh`, który wczytuje wiek podany przez użytkownika, a następnie wyświetla na ekranie komunikat stwierdzający, czy osoba jest pełnoletnia, czy nie. W przypadku podania przez użytkownika liczby ujemnej, skrypt ma wyświetlić komunikat o błędnie podanej wartości i zakończyć działanie.
3. Napisać skrypt `plik.sh`, który wczytuje nazwę pliku, a następnie jeśli w bieżącym katalogu nie ma takiego pliku ani katalogu, tworzy plik o podanej nazwie, zapisując w nim tekst "Nowy plik" i wyświetla zawartość pliku.
4. Napisać skrypt `ucz.sh`, który wczytuje podaną przez użytkownika liczbę naturalną  $n$  i następnie  $n$  razy wyświetla komunikat *Warto się uczyć*. Przy podaniu  $n$  ujemnego lub 0 powinien pojawić się odpowiedni komunikat o błędnej wartości  $n$ .
5. Napisać skrypt `liczbapierwsza.sh` który sprawdzi czy podana liczba jest liczbą pierwszą.
6. Napisać skrypt `dodaj.sh`, który wczytuje liczby całkowite podawane kolejno przez użytkownika sumując je, przy czym podanie liczby 0 kończy działanie skryptu.
7. Napisać skrypt `lista.sh`, który wczyta 10 nazwisk podanych przez użytkownika, zapisze je w pliku `lista.txt`, a następnie wyświetli zapisane nazwiska w porządku alfabetycznym.
8. Napisać skrypt `gwiazdki.sh`, który wczytuje podaną przez użytkownika liczbę naturalną  $n$  i następnie wyświetla w wierszu  $n$  gwiazdek. Przy podaniu liczby ujemnej lub 0 powinien pojawić się odpowiedni komunikat o błędnej wartości  $n$ .

9. Napisać skrypt `pliki.sh`, który wczyta podaną przez użytkownika liczbę naturalną  $n$  (przy podaniu  $n$  ujemnego lub 0 powinien pojawić się odpowiedni komunikat o błędnej wartości), a następnie w bieżącym katalogu utworzy  $n$  pustych plików o nazwach `plik1.txt`, `plik2.txt`, ..., `plikn.txt`.

10. Napisać skrypt `zlicz.sh` który policzy ilość wystąpień słowa podanego w pierwszym argumencie w pliku podanym jako drugi argument.