

Pracownia Technik Obliczeniowych bash

Paweł Daniluk

Wydział Fizyki

Wiosna 2016



Powłoka systemu

Powłoka systemu (shell) jest doskonałym narzędziem, które pozwala zrobić sporo więcej niż tylko uruchamiać programy.

Shell jest kompletnym językiem programowania.

Shell pozwala uruchamiać wiele zadań i nad nimi panować.

Zadania w tle

Każdy (?) działający program można zatrzymać (Ctrl-Z) i następnie wznowić.

Polecenia `fg`, `bg` i `jobs` pozwalają zarządzać uruchomionymi zadaniami.

Należy odróżniać zadania (`jobs`) bashowe od procesów systemu operacyjnego.

Przydatne kombinacje klawiszy:

- 1 Ctrl-C – kończy działanie programu (wysyła SIGTERM)
- 2 Ctrl-Z – zawiesza działanie zadania
- 3 Ctrl-Y – zawiesza działanie zadania w momencie, gdy oczekuje ono na standardowym wejściu
- 4 Ctrl-D – zamyka standardowe wejście zadania

Krótki slajd o zabijaniu

```
kill [-s sigspec | -n signum | -sigspec] [pid | jobspec] ...
```

```
killall [-e|--exact] [-g|--process-group] [-i|--interactive]
```

Przydatne sygnały

- (6) SIGABRT
- (9) SIGKILL
- (11) SIGSEGV
- (15) SIGTERM

Uruchamianie programów

W skrypcie bashowym można umieszczać dowolne polecenia, które bash interpretuje. Aby uruchomić program nie potrzeba nic szczególnego. Chyba, że trzeba przechwycić wyjście.

Stary sposób – odwrotne apostrofy (backticks)

```
'ls'
```

Nowy sposób

```
$(ls)
```

Pętla for

Bash ma zwykłą pętlę for-each.

```
for i in $( ls ); do
    echo item: $i
done
```

Pętli można używać w linii komend.

Pętla for

Pętlę po liczbach można zrobić korzystając z programu seq.

```
for i in `seq 1 10`;  
do  
    echo $i  
done
```

bash jest językiem programowania

Wyrażenia arytmetyczne

```
echo $(( 2 + 2 ))
```

Przypisania

```
b=$(( $a + 2 ))
```

Instrukcja warunkowa

```
if [ $a -lt 5 ]; then
    echo "a_jest_male.";
else
    echo "a_jest_spore.";
fi
```


bash jest językiem programowania

Pętla

```
while [ $k -gt 5 ]; do  
    k=$(( $k - 1 ));  
done
```

Zadanie 1

Napisz program generujący w nieskończoność kolejne liczby naturalne. Uruchom go tak, aby po wylogowaniu się i ponownym zalogowaniu nadal działał.

Zadanie 2

Stwórz katalog zawierający bardzo dużo ($>10k$) plików o nazwach będących liczbami naturalnymi. Usuń pliki o nazwach podzielnych przez 10.

Zadanie 3

Napisz skrypt, który przy pomocy `wc` zlicza linie w plikach o nazwach podanych w linii parametrów.