

mgr Maciej Wróbel

Skrypty Bash, sed, awk itd, zaawansowana obsługa Linux, wyrażenia regularne. Część I. Zadania

25 październik 2010

Zadania oznaczone * są opcjonalne.

1. Wprowadzenie – skrypty

1.1. Celowość wykorzystania języków skryptowych

*Zapoznaj się z dokumentem Scripting: Higher Level Programming for the 21st Century, dostępnym na <http://home.pacbell.net/ouster/scripting.html#924281>.

2. Podstawy pisania skryptów w Bash

2.1. Pojęcia wstępne

1. Wywołaj polecenia ls, ps i pstree jednoliniowym skrypcem.

2.2. Zmienne

1. Przypisz zmiennej mvar dowolną wartość.
2. Wyświetl wartość zmiennej mvar.
3. Utwórz plik o nazwie zawartej w mvar.

2.3. Podstawianie komend

1. Przypisz do zmiennej pliki rezultat polecenia `ls -l /`
2. Wyświetl zawartość zmiennej pliki

2.4. Dopasowywanie wzorców i podstawianie nazw plików i rozwijanie klamer

1. Zmiennej pliki przypisz pliki z katalogu /etc zaczynające się na literę x.
2. Skopiuj te pliki do katalogu domowego.
3. Utwórz plik zawierający * w nazwie.
4. Utwórz (jednym poleceniem) pliki o nazwie opisanej wzorcem plika.txt, plikb.txt, plikc.txt ..., plikg.txt.
5. Wyświetl pliki z katalogu /etc zaczynające się na litery a-f.
6. Sprawdź, jaki jest maksymalna wielkość n dla listy 1..n.

2.5. Arytmetyka powłoki

1. Sprawdź, jakie działania matematyczne można wykonać przy pomocy Bash.

2.6. Wyrażenia logiczne

1. Zapoznaj się z dokumentacją polecenia `test`.
2. Napisz polecenie, które wyświetli `ok` pod warunkiem, że katalog `/proc/5` istnieje.
3. Napisz polecenie, które utworzy katalog `abcde` pod warunkiem, że taki katalog jeszcze nie istnieje.
4. Napisz polecenie, które wyświetli (poleceniem `cat`) zawartość pliku `/var/log/user.log` pod warunkiem, że plik ten istnieje.

2.7. Rurociągi (ang. *pipelines*)

1. Zapoznaj się z poleceniami `head`, `tail`, `grep`, `sort`.
2. Podaj przykłady rurociągów z wykorzystaniem powyższych poleceń (i innych poleceń, jak `cat`, `ls`, `ps`).
3. Korzystając z poleceń `ls`, `head` (i ewentualnie `sort`) wyświetl 10 największych plików w katalogu `/bin`.

3. Programy w Bash

1. Utwórz i wykonaj w Bash program `hello world`, niezależny od powłoki w której zostanie uruchomiony.
2. Utwórz program, który przypisze dla bieżącej powłoki zmiennej `oldps1` wartość zmiennej `ps1`, a następnie ustawi wartość zmiennej `PS1` na `"buu"`. Wykonaj `go`.
3. Korzystając ze zmiennej `oldps1` przywróć stan początkowy powłoki.

3.1. Kolejność wykonywania kodu

1. Napisz program, który wyświetli *ok* jeżeli istnieje plik `/etc/passwd` oraz *niedobrze*, jeżeli takiego pliku nie ma.
2. Napisz program, który zapisze do zmiennej pliki listę plików z `/usr/bin`, a następnie w pętli wyświetli wszystkie ich nazwy.
3. Korzystając z pętli `for ... in` i polecenia `seq` napisz program, który utworzy pliki `1.txt`, `11.txt`, `21.txt`, ... `101.txt`.
4. Korzystając z pętli `for ((;))` usuń powyższe pliki (jeżeli istnieją).
5. Korzystając z polecenia `read` i pętli `while` napisz program, który będzie liczył wiersze wchodzące do standardowego wejścia.

3.2. Argumenty pozycyjne

1. Napisz program, który sprawdzi, czy wartościom kolejnych zmiennych pozycyjnych odpowiadają istniejące pliki i jeżeli nie, to wyświetli komunikat *brak pliku ...*

4. Sprawozdanie

Sprawozdanie powinno zawierać rozwiązania wszystkich zadań (oprócz tych, które polegają na czytaniu dokumentacji). Dodatkowo opisz 10 wybranych wbudowanych poleceń Bash (podręcznik Bash, builtin commands) oraz streść jeden, Twoim zdaniem najciekawszy rozdział podręcznika Bash.