

Powłoka BASH

Autor: Administrator
01.04.2006.

Po uruchomieniu Linuksa w powłoce tekstowej widzimy anemiczny czarny monitor z białymi literkami. To prawdziwy Linux. Musimy działać za pomocą klawiatury wpisując polecenia.

POWŁOKA czyli shell to program, który przyjmuje polecenia człowieka i uruchamia inne programy. Wybór rodzaju shella dla danego usera jest zawarty w pliku /etc/passwd. Przyjmujemy, że pracujemy na shellu bash. Podajemy swój login np. antek i po wciśnięciu ENTERa podajemy hasło (plus ENTER). Jesteśmy zalogowani jako antek.

Jest tak wiele rzeczy do wyjaśnienia na początku, że proszę o wyrozumiałość jeżeli nie trafiłem z hierarchią ważności.

Uruchamianie programów w shellu może odbywać się na dwa sposoby: bez podania PEŁNEJ ŚCIEŻKI DOSTĘPU do pliku inicjującego lub ze ścieżką. Odpowiedzmy sobie: jak to się dzieje, że shell czasami zna PEŁNĄ ŚCIEŻKĘ, a czasami nie? Otóż dawno, dawno temu, ktoś wpadł na pomysł, by zamiast za każdym razem wpisywać pełną ścieżkę dostępu do pliku inicjującego (aby uruchomić program) - podawać tylko nazwę pliku, a wszystkie pliki inicjujące UMIESZCZAĆ W JEDNYM lub kilku ściśle wskazanych katalogach. W takich okolicznościach shell mając zlecenie zbudowane tylko z nazwy pliku, szuka pliku inicjującego o tej samej nazwie w wybranych katalogach wskazanych poprzez zmienną PATH.

PATH czyli ŚCIEŻKI PRZESZUKIWANIA. Domyślnie umieszczamy pliki inicjujące w katalogach: /bin , /usr/local/bin, gdyż tam shell odnajdzie je bez problemu. O położeniu tych katalogów został poinformowany podczas startu systemu odczytując dane zmiennej PATH. Jak dodawać swoje ścieżki wyszukiwania? Można na dwa sposoby:

Na czas jednej sesji (do restartu systemu lub wylogowania usera) dodajemy nowy katalog /home/antek/bin do zmiennej PATH zleceniem (jako zwykły user):

```
$ export PATH="$PATH:~/bin" . Wyjaśnienie: znaczek dolara $ oznacza zwykłego usera np. antek (służy do opisu, $ nie wpisuje się w zlecenie), tylda ~ oznacza skrót do katalogu domowego usera czyli /home/antek . Aby wyświetlić listę ścieżek przeszukiwania i sprawdzić, czy nasz wpis zadziałał - daj zlecenie:
```

```
$ echo $PATH
```

Aby ustawienia zmiennej PATH włączały się podczas startu systemu, należy dopisać treść zlecenia export PATH="\$PATH:~/bin" lub PATH="\$PATH:~/bin" do pliku:

```
~/ .bash_profile (w Mandraku ten wpis już jest ;)  
lub jeżeli go nie ma, to ~/bash_login  
lub jeżeli ich nie ma to ~/.profile
```

Wyjaśnienie: kropka na początku nazwy pliku oznacza, że ma artybut UKRYTY. Warto w celach edukacyjnych przejrzeć plik ~/ bashrc oraz .bash_history z zapisem historii wydawanych poleceń. Kilka słów o .bash_history tutaj.

Zmienną środowiska PATH można także ustalić na stałe poprzez modyfikację pliku /etc/rc.d/rc.sysinit Tak na marginesie informacja: jako pierwszy (podczas konfiguracji basha) jest czytany plik /etc/profile, a następnie wymienione powyżej w pkt. b. Natomiast podczas wylogowania jest odczytywany plik ~/ bash_logout . W katalogu /etc znajduje się jeszcze plik bashrc. Do niego odwołują się pewne wpisy w pliku ~/bashrc zwykłych userów.

Powyższe wzory plików konfiguracyjnych basha są pobrane z mojego Mandraka 9.0.

Jakie są praktyczne zastosowania zmiennej PATH? Przypuśćmy, że w naszym Linuksie mamy program KADU działający w powłoce tekstowej. Jeżeli został zainstalowany w /home/antek/bin/.gg to uruchamianie programu będzie wymagało podania pełnej ścieżki do pliku w katalogu /home/antek/bin/.gg . Ułatwimy sobie pracę i po ustaleniu dodatkowej ścieżki w zmiennej PATH wystarczy, że jako antek wpisujemy zlecenie gg (o ile gg jest nazwą pliku inicjującego program).

Przypominam, że aby wyświetlić aktualną listę ścieżek przeszukiwania, daj zlecenie:

```
$ echo $PATH
```

Ponieważ jesteś (jak przypuszczam) początkującym miłośnikiem Linuksa poznaj inny, wygodny sposób uruchamiania programów bez konieczności ustawiania zmiennej PATH. Utwórz skrypcik zleceniem touch nazwaskryptu, a następnie nadaj mu właściwe prawa zleceniem: chmod 751 nazwaskryptu .

Wyedytuj plik za pomocą jakiegoś edytorka (np. mcedit) i wprowadź poniższą zawartość:

```
#!/bin/sh
```

```
cd /home/antek/bin/.gg;./gg
```

Skrypt ma w sobie dwa polecenia: przeniesienie do katalogu z plikiem gg, a następnie uruchomienie pliku gg. Aby uruchomić skrypcik, trzeba być w katalogu gdzie jest ten skrypt i wydać zlecenie z przedrostkiem ./ np. ./nazwaskryptu

Poprawianie wyglądu shella podczas logowania załatwi nam wpis w pliku ~/.bashrc (który dodatkowo może współpracować z innymi plikami konfiguracyjnymi np. /etc/bashrc). Mamy dwa sposoby wprowadzania korekt:

korekta przy pomocy znaków specjalnych zmiennej PS1

korekta przy pomocy zewnętrznego skryptu odpalanego za pomocą wpisu w ~/.bashrc

Proszę przyjrzeć się poniższemu plikowi ~/.bashrc (czyli w praktyce /home/antek/.bashrc - kropka oznacza, że plik ma atrybut "ukryty").

```
# .bashrc
```

```
# User specific aliases and functions
```

```
# Source global definitions
```

```
if [ -f /etc/bashrc ]; then
```

```
./etc/bashrc
```

```
fi
```

```
PS1='ANTEK $PWD>'
```

```
./powitanie
```

Nas interesują miejsca:

```
if [ -f /etc/bashrc ]; then - co oznacza, że domyślnie dane są pobierane z ogólnego pliku /etc/bashrc
```

PS1='\$PWD>' - dopisany przeze mnie (na końcu pliku) wiersz zmuszający znak zachęty do umieszczania wyniku zlecenia pwd (aktualne położenie usera) lub PS1='ANTEK \$PWD>' czyli wiersz zmuszający znak zachęty do dodatkowego wydrukowania na ekranie słowa ANTEK (oraz umieszczania wyniku zlecenia pwd). Uwaga: zamiast słowa ANTEK, można użyć znak specjalny wprowadzający nazwę usera. Na sąsiedniej stronie zrobiłem listę znaków specjalnych zmiennej PS1.

Dzięki dopisaniu w pliku ~/bashrc wiersza PS1='ANTEK \$PWD>' uzyskamy bardzo wygodny znak zachęty, wyświetlany przez cały czas pracy w shellu. Uwaga: słowo ANTEK będzie wyświetlane także po zalogowaniu się na roota (za pomocą zlecenia su). Dlatego polecam w /root/bashrc dopisanie (na końcu pliku) podobnego wiersza PS1='ROOT \$PWD>'. Przykład pliku /root/bashrc

./powitanie - czyli odpalenie skryptu powitanie . Uruchomienie skryptu zawdzięczamy prostemu wpisowi

./powitanie w pliku ~/bashrc (oczywiście dla ułatwienia oba pliki są w tym samym katalogu).

Przypominam, że skrypt ten powinien mieć prawo do wykonania czyli np. 644 antek.antek . Uwaga: jeżeli zamontujesz partycję /home w trybie noexec (patrz plik /etc/fstab), to system nie zdoła uruchomić skryptu powitanie i nie będzie komunikatu :(

Budowa skryptu powitanie:

```
#!/bin/sh#czyli konfiguracja shella dla skryptu
```

```
echo 'JESTES NA KOMPUTERZE: '#czyli wywietlenie komunikatu JESTEŚ...
```

```
cd /bin;./hostname#czyli uruchomienie zlecenia hostname (wyświetli się nazwa naszego komputera)
```

```
echo '#czyli jeden pusty wiersz
```

```
echo 'DZISIAJ MAMY: '
```

```
cd /bin;./date
```

```
echo ''
```

```
echo 'sprawdzam poczte... '
```

```
mail
```

```
echo ''
```

Proszę zwrócić uwagę na znaczek ' (jest na klawiszu z cudzysłowiem). Wpis echo '' daje nam wolny wiersz.

Proszę zobaczyć jakie komunikaty daje zlecenie uname : nazwa systemu, nazwa hosta, ostatnie logowania.

Przypominam, że wypełniając plik /etc/motd jakimś tekstem - uzyskasz komunikat powitalny (po zalogowaniu). Jeżeli chcesz usunąć możliwość wyświetlenia informacji o systemie w pliku /etc/motd, to zahaszuj w pliku /etc/rc.d/rc.S wiersz:

```
echo "$(/bin/uname -sr)." > /etc/motd
```

Wymuszone wylogowanie po ustalonym okresie bezczynności załatwi wpis w pliku ~/.bashrc

```
export TMOU=300
```

wartość 300 oznacza czas bezczynności (ilość sekund) po jakim nastąpi wylogowanie

Jeżeli nie chcesz modyfikować plików konfiguracyjnych, to możesz po prostu wpisać do konsoli zlecenie:

```
export TMOU=300
```

Ustawianie parametrów shella u nowego usera załatwia zawartość katalogu /etc/skel . To z tamtą

zlecenie `useradd` kopiuje pliki i umieszcza w domowym katalogu nowego usera `/home/user`. W katalogu `/etc/skel` są: `.bash_logout`, `.bash_profile`, `.bashrc`, `.mailcap`, `.screenrc`

Jeżeli męczy Ciebie nadmierna ilość systemowych komunikatów ostrzegawczych (w powłoce tekstowej), wyedytuj plik `/etc/syslog.conf` i usuń wpis:

*.emerg root (komunikaty pojawią się na shellu roota)

*.emerg * (komunikaty pojawią się na shell każdego usera)

Ang. emergency czyli pilne są ważnym elementem bezpieczeństwa, więc przemyśl usunięcie wpisu.

Więcej o `syslog` znajdziesz na stronie pt. BHPoraz ZARZĄDZANIE.

Podczas kompilacji instalowanego programu możesz spotkać się z problemem nieumiejętności odnalezienia bibliotek położonych w nietypowych katalogach. W takich okolicznościach zwiększ liczbę katalogów, które przeszukiwane są podczas kompilacji. Jako root wyedytuj plik `/etc/ld.so.conf` zleceniem `mcedit ld.so.conf` i dopisz nowe katalogi. Po zapisaniu zmian daj zlecenie `ldconfig`

Tworzenie skróconych wersji często używanych, rozbudowanych poleceń. Aliasy to mechanizm powłoki, który umożliwia tworzenie skróconych wersji poleceń. Alias tworzymy wpisując w powłoce słowo alias, wygląd "skróconej wersji zlecenia", znak równości=, zlecenie oryginalne (np. z parametrami) w apostrofach, czyli:

`alias ll='ls -la'` wprowadzający skróconą wersję (`ll`) listowania katalogu (`ls -la`). Taki alias ma krótką trwałość życia - do czasu wylogowania się.

Samo zlecenie `alias` (bez argumentów) wyświetla aktualną listę aliasów.

Zlecenie `unalias nazwaaliasu` usuwa dany alias, natomiast `unalias -a` usuwa wszystkie aliasy.

Aby trwale skompletować i uruchamiać aliasy podczas logowania - należy utworzyć zwykły plik tekstowy pod nazwą `/home/user/.alias` i wpisać wiersze z aliasami. Następnie należy wpisać do `/home/user/.bashrc` wiersz: `./alias`. Tyle mówią podręczniki - mi nie udało się uruchomić w ten sposób aliasów...

źródło <http://gorzow-wlkp.pl/>