

Dodatek nr 1: Niektóre polecenia systemu Linux (shell tcsh)

A) Informacje wstępne

Polecenia w terminalu systemu **Linux** podaje się poprzez wprowadzenie nazwy polecenia i naciśnięcie klawisza **Enter**. Możliwe jest również uruchomienie polecenia w tle – aby to wykonać polecenie należy zakończyć znakiem **&** (z przerwą lub bez przerwy między poleceniem za znakiem **&**). Uruchomienie w tle spowoduje „zwolnienie” terminala i można w nim wykonywać kolejne polecenia. Przykład uruchomienia graficznego edytora tekstu w nowym oknie i zwolnienie terminala do innych prac:

```
gedit &
```

W przypadku poleceń wykonywanych natychmiastowo dodawanie znaku **&** nie ma sensu bo terminal i tak zostaje zwolniony po zakończeniu wykonywania polecenia, jakkolwiek nie jest to błędem.

Uwaga: system *Linux* jest czuły na wielkość liter zarówno jeśli chodzi o wprowadzane polecenia jak i nazwy plików!

W przypadku używania powłoki (ang. shell) tcsh możliwe są pewne wbudowane udogodnienia:

- naciśnięcie klawisza **TAB** w czasie wprowadzania nazw plików i katalogów wykonuje autozakończanie nazw plików, jeśli w danym katalogu jest kilka możliwości zakończenia nazwy są one wylistowane w celu pomocy wyboru operatorowi, np. napisanie w terminalu litery **a** i naciśnięcie klawisza **TAB** wylistuje wszystkie pliki zaczynające się literą **a** lub jeśli będzie tylko jeden taki plik spowoduje wypisanie jego nazwy,
- naciskanie strzałek góra/dół przywołuje ostatnio wykonane polecenia i można je łatwo powtórzyć bez konieczności ponownego ich wpisywania,
- zaznaczenie tekstu występującego w terminalu poprzez użycie lewego klawisza myszki powoduje automatyczne skopiowanie tego tekstu do schowka, naciśnięcie prawym klawiszem myszki w danym miejscu powoduje automatyczne wklejenie tekstu ze schowka.

Rodzaj używanej powłoki można sprawdzić poprzez wydanie polecenia:

```
ps
```

W odpowiedzi uzyskamy wszystkie procesy pracujące w tle. W przypadku świeżo otworzonego terminala wynik może być jak poniżej umieszczony:

```
[s_test@pglaser2 ~]$ ps
  PID TTY          TIME CMD
 19519 pts/9        00:00:00 tcsh
 19727 pts/9        00:00:00 ps
```

Obecność procesu o nazwie **tcsh** świadczy o używaniu właśnie tej powłoki. Polecenia wydawane w terminalu mają swój system pomocy, który możemy uruchomić poprzez podanie nazwy danego polecenia z parametrem wywołania **--help**. Przykład wywołania pomocy dla polecenia **ps** przedstawiony jest poniżej:

```
ps --help
```

Poniżej umieszczono fragment pomocy uzyskanej z powyższego polecenia:

```
[s_test@pglaser2 ~]$ ps --help
***** simple selection *****      ***** selection by list *****
-A all processes                      -C by command name
-N negate selection                   -G by real group ID (supports names)
```

Pomoc dotyczącą systemu można także uzyskać za pomocą polecenia **man**. Przykład użycia polecenia do odszukania pomocy dotyczącej polecenia **more**:

```
man more
```

Przewijanie pomocy uzyskuje się klawiszami strzałek góra/dół i spacji a wyjście z polecenia **man** uzyskuje się poprzez naciśnięcie klawisza **"q"**.

B) Operacje na katalogach i plikach

System plików w systemie Linux nie ma tak zwanych napędów oznaczanych kolejnymi literami a będących zazwyczaj odpowiednikiem fizycznego urządzenia takiego jak np. HDD, CDROM/DVD, FDD lub napęd USB znanego w systemie Windows. Zamiast tego system plików jest zorganizowany jako drzewo zaczynające się od korzenia (oznaczanego znakiem ukośnika „/”) a poszczególne napędy zmapowane są do podkatalogów. Najczęściej główny dysk komputera (jedna z jego partycji lub też zasób złożony z kilku dysków) zmapowana jest do korzenia /. Kolejne dyski, jeśli występują, są zazwyczaj zmapowane do jakiegoś podkatalogu np. do katalogu `/mnt`. Katalogi domowe użytkowników w systemie z domyślnymi ustawieniami są zazwyczaj umieszczone jako podkatalogi w folderze `/home`.

Uwaga: na komputerach w sieci KSMI katalogi domowe użytkowników są umieszczone w katalogu `/home2`.

Każdy użytkownik ma pełen dostęp do swojego katalogu domowego. Poza tym katalogiem użytkownik ma zazwyczaj dostęp tylko do odczytu do wybranych plików systemowych. Dany użytkownik, będący równocześnie właścicielem swojego katalogu domowego może zmienić prawa dostępu do własnych plików i np. umożliwić dostęp do nich innym użytkownikom. Poniżej przedstawione zostaną podstawowe polecenia służące do pracy na plikach i katalogach. W poniższych przykładach założono, że katalog domowy jest skonfigurowany jako `/home2` a do systemu zalogował się użytkownik o loginie `s_test`. Wyświetlenie bieżącego katalogu:

```
pwd
```

Zmiana katalogu na katalog domowy:

```
cd
```

W wyniku polecenia `cd` katalog zostanie zmieniony na `/home2/s_test`.

Zmiana katalogu na katalog `/etc` (jest to katalog systemowy do którego użytkownicy mają zazwyczaj dostęp tylko do odczytu):

```
cd /etc
```

Wyświetlenie zawartości katalogu bieżącego:

```
ls
```

W wyniku wykonania powyższego polecenia w terminalu są wylistowane nazwy plików i katalogów umieszczonych w bieżącym katalogu.

Wyświetlenie zawartości katalogu bieżącego wraz ze szczegółowymi informacjami i plikami ukrytymi:

```
ls -la
```

W wyniku wykonania powyższego polecenia w terminalu są wylistowane nazwy plików wraz z: prawami dostępu, właścicielem, grupą, wielkością i datą ostatniej modyfikacji.

Symbol kropki `..` oznacza w Linuxie katalog bieżący a dwie kropki `..` oznaczają katalog nadrzędny, stąd zmiana o jeden katalog wyżej będzie możliwa poprzez podanie polecenia:

```
cd ..
```

a wyświetlenie zawartości bieżącego katalogu może być wykonane przez:

```
ls
```

lub równoznacznie przez:

```
ls .
```

lub też poleceniem

```
ls ./
```

Tworzenie nowego katalogu (będącego podkatalogiem w bieżącym katalogu) wykonuje się przez:

```
mkdir nazwa_nowego_katalogu
```

Kopiowanie plików wykonuje się poprzez polecenie **cp**, można przy tym stosować znak gwiazdki "*" oznaczający dowolny ciąg znaków i znak zapytania "?" oznaczający dowolny pojedynczy znak. Poniżej zamieszczono kilka przykładów:

Skopiowanie pliku **a.txt** z bieżącego katalogu do katalogu o nazwie **podkatalog**:

```
cp a.txt podkatalog
```

W powyższym przykładzie katalog o nazwie **podkatalog** musi istnieć wcześniej niż wydane polecenie kopiowania, w przeciwnym razie plik o nazwie **a.txt** zostanie skopiowany do nowego pliku o nazwie **podkatalog**.

Skopiowanie wszystkich plików których nazwa rozpoczyna się literą **a** i kończy **.txt** (pomiędzy **a** i **.txt** może występować dowolny ciąg znaków) do katalogu o nazwie **podkatalog**:

```
cp a*.txt podkatalog
```

Kopiowanie tworzy nowy plik o identycznej zawartości w miejscu docelowym pozostawiając stary plik w miejscu źródłowym. Jeśli chcemy aby stary plik nie pozostawał w systemie możemy go skasować albo do kopiowania użyć polecenia **mv**. Polecenie **mv** działa identycznie jak **cp** nie pozostawiając jednak oryginału w miejscu źródłowym.

Kasowanie plików wykonuje się poprzez polecenie **rm**. Przykład, skasowanie pliku **a.txt** w podkatalogu o nazwie **podkatalog**:

```
rm ./podkatalog/a.txt
```

Przykład, skasowanie podkatalogu o nazwie **podkatalog** wraz z jego całą zawartością (uwaga: polecenie z opcją **-rf** należy stosować bardzo ostrożnie bo można łatwo skasować ważne dane):

```
rm -rf podkatalog
```

Zmianę właściciela do pliku/katalogu wykonuje się poprzez wykorzystanie polecenia **chown** a zmianę praw dostępu do plików poprzez polecenie **chmod**.

C) Operacje na zmiennych środowiskowych

Na wstępie tego podrozdziału należy wyraźnie zaznaczyć, że poniższe polecenia odnoszą się wyłącznie do powłoki tesh. Do powołania i ustalenia wartości zmiennej środowiskowej służy polecenie **setenv**. Zmienne środowiskowe mogą przy tym być powołane do życia bez posiadania jakiegokolwiek wartości.

Powołanie zmiennej środowiskowej o nazwie **ABC** bez nadawania wartości:

```
setenv ABC
```

Usunięcie zmiennej środowiskowej **ABC**:

```
unset ABC
```

Powołanie zmiennej środowiskowej o nazwie **ABC** z nadaniem wartości równej **value_abc**:

```
setenv ABC value_abc
```

Odczyt wartości zmiennej środowiskowej dokonywany jest poprzez podanie znaku dolara **\$** przed nazwą zmiennej i tak dla polecenia:

```
echo $ABC
```

W wyniku wykonania powyższych poleceń powinniśmy uzyskać następujący wynik:

```
[s_test@pglaser2 ~]$ echo $ABC  
value_abc
```

Niektóre zmienne środowiskowe mają szczególne znaczenie i ich prawidłowe skonfigurowanie jest bardzo ważne dla poprawnej pracy systemu. Polecenie **setenv** bez podania parametrów listuje wszystkie aktualnie istniejące zmienne środowiskowe wraz z ich wartościami:

```
setenv
```

Skrócony wynik powyższej operacji przedstawiony jest tu:

```
[s_testp@pglaser2 ~]$ setenv
USER=s_test
HOME=/home2/s_test
PATH=/usr/lib64/qt-3.3/bin:/usr/local/bin:/bin:/usr/bin
HOSTTYPE=x86_64-linux
VENDOR=unknown
OSTYPE=linux
PWD=/home2/bpa_ldap
GROUP=staff
HOST=pglaser2
REMOTEHOST=bm2.ue.eti.pg.gda.pl
HOSTNAME=pglaser2
LS_COLORS=rs=0:di=01;
CVS_RSH=ssh
G_BROKEN_FILENAMES=1
SSH_ASKPASS=/usr/libexec/openssh/gnome-ssh-askpass
LANG=en_US.UTF-8
ABC=value_abc
```

Jak widać powyżej, w zmiennych środowiskowych zapisywanych jest wiele istotnych dla bieżącej pracy parametrów. Zmienna **PATH** wyszczególnia wszystkie katalogi w których będzie wyszukiwane polecenie wpisane w terminalu i zlecone do wykonania poprzez naciśnięcie klawisza **Enter**.

D) Pozostałe przydatne programy i polecenia.

Zmiana hasła dostępu do konta:

```
passwd
```

Uruchomienie graficznego edytora tekstowego **gedit** w tle wraz z utworzeniem pliku **a.txt** w bieżącym katalogu:

```
gedit a.txt &
```

Wyświetlenie zawartości pliku tekstowego **a.txt** w terminalu:

```
more a.txt
```

lub

```
less a.txt
```

lub

```
cut a.txt
```

Uruchomienie programu analizującego wykorzystanie zasobów (wyjście z programu – klawisz **q**):

```
top
```

Znalezienie miejsca z którego wykonywane jest dane polecenie: **which**. Przykład poszukiwania ścieżki do pliku wykonywalnego edytora **gedit**:

```
which gedit
```

daje wynik w postaci:

```
[s_test@pglaser2 ~]$ which gedit
/usr/bin/gedit
```

Poszukiwania plików dokonuje się przy użyciu polecenia **find**. Składnia polecenia jest dość złożona więc przedstawiony zostanie wyłącznie jeden przykład polegający na próbie odnalezienia pliku o nazwie **fstab** w katalogu **/etc**:

```
find /etc -name "fstab"
```

Wyniki wyszukiwania przedstawione są poniżej i przedstawiają próby wejścia do każdego z podkatalogów folderu **/etc** wraz z niepowodzeniami (bo dostęp jest zabroniony przez system). Odnaleziony został jeden plik w położeniu **/etc/fstab**.

```
[s_test@pglaser2 ~]$ find /etc -name "fstab"
find: `/etc/ntp/crypto': Permission denied
find: `/etc/audisp': Permission denied
find: `/etc/pki/rsyslog': Permission denied
find: `/etc/pki/CA/private': Permission denied
find: `/etc/cups/ssl': Permission denied
find: `/etc/sss': Permission denied
/etc/fstab
find: `/etc/dhcp': Permission denied
find: `/etc/selinux/targeted/modules/active': Permission denied
find: `/etc/audit': Permission denied
find: `/etc/sudoers.d': Permission denied
find: `/etc/polkit-1/localauthority': Permission denied
find: `/etc/lvm/cache': Permission denied
find: `/etc/lvm/backup': Permission denied
find: `/etc/lvm/archive': Permission denied
```

Uruchomienie menedżera plików **Midnight Commander**:

```
mc
```

Uruchomienie w tle przeglądarki internetowej **Mozilla Firefox**:

```
firefox &
```

Uruchomienie w tle przeglądarki internetowej **Mozilla Firefox** wraz z jednoczesnym wczytaniem wszystkich plików typu **pdf** z katalogu bieżącego:

```
firefox *.pdf &
```

Spakowanie całego katalogu o nazwie **podkatalog** do jednego pliku o nazwie **podkatalog.tar.gz** (poleceniem **tar**) i jego skompresowanie poleceniem **gzip**:

```
tar -zcvf podkatalog.tar.gz podkatalog
```

Czasami trzeba wykonać archiwizację całego katalogu domowego (w przykładzie poniżej o nazwie **../bpa_ldap**) - w tym celu można wykonać polecenie:

```
tar -zcvf home_dir.tar.gz --exclude=home_dir.tar.gz ../bpa_ldap
```

które wykona utworzenie archiwum o nazwie **home_dir.tar.gz** i do niego skopiuje wszystkie pliki z katalogu bieżącego oraz podkatalogów (z wyłączeniem samego nowo tworzonego archiwum).

Rozpakowanie spakowanego i skompresowanego archiwum o nazwie **podkatalog.tar.gz** do bieżącego katalogu:

```
tar -zxvf podkatalog.tar.gz
```

lub do dowolnego innego katalogu:

```
tar -zxvf podkatalog.tar.gz -C /home2/s_test/katalog_docelowy/
```

Zdalne kopiowanie plików pomiędzy komputerami z systemem Linux pracującymi w sieci możliwe jest poprzez polecenie **scp**. W poniższym przykładzie użytkownik **s_test** (musi on posiadać konta na obu komputerach **fpgalab12** i **pglaser2**) wykonuje kopię katalogu **folder_zrodlowy** wraz ze wszystkimi plikami i podkatalogami z komputera **fpgalab12** na komputer **pglaser2** do katalogu **folder_docelowy**. Zarówno **folder_zrodlowy** jak i **folder_docelowy** umieszczone są w katalogach domowych a użytkownik **s_test** jest obecnie zalogowany do komputera **fpgalab12**:

```
scp -r folder_zrodlowy s_test@pglaser2:/home2/s_test/folder_docelowy
```

Sprawdzenie wykorzystania systemu dyskowego:

df -h

Sprawdzenie zajętości bieżącego katalogu (wraz z podkatalogami):

du -sh .

Sprawdzenie zajętości z wyszczególnieniem zajętości dla poszczególnych podkatalogów:

dh -h .