

Korzystanie z terminala

Po uruchomieniu terminala użytkownik uzyskuje tzw. znak zachęty (ang. shell prompt), np. \$ lub #. Po zobaczeniu znaku zachęty można już zacząć wpisywać polecenia dla systemu.

Historia wykonanych poleceń zapisywana jest w katalogu domowym użytkownika w pliku .bash_history. Aby przywołać użyte ostatnio polecenia podczas korzystania z terminala, należy użyć klawiszy , 

Warto pamiętać o opcji autouzupełniania, której używamy naciskając klawisz [Tab]

Aby przerwać działanie polecenia, należy użyć kombinacji klawiszy [Ctrl]+[C]

Znaki globalne

* reprezentuje jeden lub wiele znaków

? reprezentuje dokładnie jeden znak

[abcde] reprezentuje dokładnie jeden znak z wymienionych

[a-e] reprezentuje dokładnie jeden znak z przedziału

[!abcde] reprezentuje dowolny niewymieniony znak

[!a-e] reprezentuje znak nienależący do przedziału

{plik,file} reprezentuje dowolny z wymienionych ciągów

Przykłady użycia znaków globalnych:

* reprezentuje wszystkie pliki

*.bmp reprezentuje wszystkie pliki kończące się na .bmp

?[0-9][a-d] reprezentuje plik o trzysznakowej nazwie, gdzie pierwszy znak jest dowolny, drugi jest cyfrą, zaś trzeci literą z przedziału a-d

{nowy,new}*.txt reprezentuje wszystkie pliki zaczynające się od nowy lub new będące plikami txt

Podręcznik systemowy

(ang. manual) najczęściej instalowany wraz z systemem, zawiera opisy poszczególnych poleceń systemowych. Do przewijania tekstu używamy klawiszy: ,  opis opuszczamy naciskając klawisz [Q]. Nową, ulepszoną wersją podręcznika man jest info.

Dostępna jest także skrócona wersja pomocy po wpisaniu polecenia: -help

Aby wywołać podręcznik dla np. polecenia ls możemy użyć następującej składni:

man ls

info ls

ls -help

Poruszanie się między katalogami ułatwiają specjalne oznaczenia:

~ katalog domowy zalogowanego użytkownika

/ katalog główny

. katalog bieżący

.. katalog bezpośrednio nadrzędny

Symboli tych można używać bezpośrednio w ścieżkach dostępu, np.:

cd ~/Publiczny/folder

touch ~/Dokumenty/plik.txt

Polecenia związane z wielodostępnością:

whoami (ang. kim jestem) podaje identyfikator konta, z którego aktualnie się korzysta

who (ang. kto) wyświetla listę aktualnie zalogowanych w systemie użytkowników

su (od ang. switch user) pozwala zmienić użytkownika, to znaczy zalogować się na inne konto w systemie

Polecenia związane z wielozadaniowością:

ps (od ang. processes) wyświetla listę procesów w systemie

kill (ang. zabij) usuwa w bezpieczny sposób proces o podanym numerze PID („bezpieczny” oznacza tutaj zwolnienie całego obszaru pamięci zajmowanego przez proces oraz zamknięcie plików otwartych dla procesu)

Polecenia związane z zarządzaniem plikami i katalogami:

cd (od ang. change directory) zmienia bieżący katalog roboczy

pwd (od ang. print working directory) wyświetla ścieżkę dostępu do bieżącego katalogu roboczego

find (ang. znajdź) wyszukuje pliki

Dowiązania. Rodzaje dowiązań

i-węzeł (ang. index-node) struktura danych, która pozwala systemowi Linux przechowywać informacje o plikach i katalogach. Każdy węzeł jest skojarzony z jednym zbiorem (np. plikiem lub folderem) i ma unikatowy numer, który pozwala na jego identyfikację

Dowiązanie twarde (ang. hard link) umożliwia odwoływanie się do jednego zbioru danych za pomocą różnych nazw (referencji). Przypisuje podaną nazwę do i-węzła danego zbioru danych.

Aby można było utworzyć referencję dany plik/folder musi istnieć na dysku

Dowiązanie symboliczne (ang. symbolic link) tworzy wskazanie na nazwę danego pliku, która to nazwa dopiero jest powiązana z i-node. Odpowiednikiem dowiązania symbolicznego w Windowsie jest skrót

Polecenia związane z tworzeniem dowiązań:

ln (od ang. link - połączenie) tworzy dowiązanie twarde

ln -s (od ang. symbolic link) tworzy dowiązanie symboliczne

Wyszukiwanie plików w systemie Linux realizujemy przy użyciu polecenia find.

Składnia tego polecenia jest następująca: find przeszukiwany_zbiór warunek parametr_dla_warunku.

Przykładowo poleceniem find /home/uczen -name *.txt wyszukamy wszystkie pliki, których nazwa kończy się wyrażeniem .txt znajdujące się w katalogu /home/uczen i jego podfolderach.

Warunki często używane z poleceniem find:

name - nazwa szukanego pliku

type - typ szukanego pliku

maxdepth - rekursywne szukanie w podkatalogach do określonej głębokości

mindepth - rekursywne szukanie w podkatalogach od określonej głębokości

size - szukanie pliku według jego rozmiaru

cmin - szukanie pliku według czasu modyfikacji

exec - pozwala wykonać dodatkowe polecenie na wynikach wyszukiwania

perm - szukanie plików mających określone uprawnienia

empty - szukanie pustych plików lub katalogów

Przydatnym poleceniem przy wyszukiwaniu plików może być komenda locate.

Tworzy ona bazę danych z nazwami i lokalizacją wszystkich plików w systemie.

Baza ta znajduje się w pliku /var/lib/mlocate/mlocate.db.

Przykład użycia: locate *.log - wyszukuje w systemie wszystkie pliki kończące się na .log.

Informacje o bazie otrzymamy używając parametru -S.

Jeśli chcemy przeszukiwać treść plików, to najlepszym poleceniem do tego jest grep.

Podstawowe użycie: grep "igła" stóg_siana - wyszukaj wystąpienia słowa igła w pliku stóg_siana.

Przydatne parametry tego polecenia:

i - szukaj nie zwracając uwagi na wielkość liter

w - szukaj tylko całych wyrazów

A n – pokaż n linii po znalezionym słowie

B n- pokaż n linii przed znalezionym słowem

C n - pokaż n linii wokół znalezionego słowa

c - wyświetl liczbę linii, gdzie znaleziono tekst

n - wyświetl numer linii, gdzie znaleziono tekst

Polecenie grep jest również przydatne w połączeniu z poleceniem cat.

Przykład użycia: cat /proc/cpuinfo | grep model

Znaki specjalne

| - potok, standardowe wyjście jest kierowane na standardowe wejście (np.: `grep passwd | more`)

< - przekierowanie wejścia (np.: `patch < łata`)

> - przekierowanie wyjścia (np.: `echo jaki tekst > jakiś_plik`)

>> - dołączenie przekierowanego wyjścia (np.: `cat >> jakiś_plik`)

& - operator sterujący, polecenie wykonywane w tle (np.: `mpg123 /mp3/* &`)

;- separator (np.: `mkdir katalog; cd katalog`)

<Tab> - dopisuje komendę lub ścieżkę

<Shift> - przesuwa ekran do góry

\ - nie interpretuje następnego znaku (\ENTER - kontynuuje polecenie w następnej linii)

&& - jeżeli pierwsze polecenie powiodło się zrób drugie (np.: `mkdir katalog && cd katalog`)

&> - przekierowanie standardowego wyjścia i wyjścia błędów

2> - przekierowanie wyjścia błędów

* - dopasowuje dowolny ciąg znaków

? - dopasowuje pojedynczy znak

. - katalog, w którym się znajdujesz

.. - katalog nadrzędny względem tego w którym obecnie jesteś

~ - katalog domowy

Operacje na plikach i katalogach w Linux.

Polecenia związane z zarządzaniem plikami i katalogami:

`touch`

(ang. dotknięcie — nazwa wzięła się od Unixa, gdzie polecenie to zmieniałoby datę ostatniej modyfikacji pliku na obecną) tworzy plik

`rename` (ang. zmień nazwę) zmienia nazwę wielu plików

`cat` (ang. to concatenate - łączyć) wyświetla zawartość pliku (np. tekstowego), może też połączyć dwa pliki w jeden

Polecenia związane z plikami i katalogami:

-Polecenia związane z katalogami:

--ls -Przeznaczenie: sprawdzanie zawartości katalogu

-Parametry: `ls [parametr] [katalog]`

`ls` - zostaną wyświetlone pliki i katalogi jeden po drugim

ls -a - zostaną wyświetlone wszystkie pliki (ukryte) i katalogi

ls -l - zostaną wyświetlone pliki i katalogi w postaci listy

-ls -t - zostaną wyświetlone pliki i katalogi według daty, najpierw wyświetlane są pliki nowsze

ls -r - zostaną wyświetlone pliki i katalogi według daty, najpierw wyświetlane są pliki starsze

-Opis: jest możliwość łączenia parametrów w postaci ls -al jednak źle będą rozpoznawane ls -a -l

--dir -Przeznaczenie: sprawdzanie zawartości katalogu

-Opis: to samo po napisaniu ls (bez parametrów)

--pwd -Przeznaczenie: sprawdzanie, w którym jesteśmy katalogu

-Opis: po wpisaniu komendy pokazuje się odpowiedź

--cd -Przeznaczenie: zmiana katalogu

-Parametry: ls [parametr lub katalog]

cd .. - cofamy się o 1 katalog na dół

cd ../.. - cofamy się o 2 katalogi na dół

cd gry - otwieramy katalog gry

-cd gry/tetris - otwieramy katalog tetris w katalogu gry

cd ../programy - otwieramy katalog programy w katalogu niżej

cd /programy - otwieramy katalog programy na samym "dole"

cd - otwieramy własny katalog

-Opis: jest to możliwość zmieniania aktualnego katalogu

--rmdir -Przeznaczenie: usuwanie katalogu

-Parametry: rmdir [katalog(i)]

rmdir kat1 - usunięcie katalogu kat1

rmdir kat1 kat2 - usunięcie katalogu kat1 i kat2

rmdir kat1/kat10 - usunięcie katalogu kat10 w katalogu kat1

-Opis: jest możliwość usunięcia katalogu, UWAGA katalog musi być pusty

--mkdir -Przeznaczenie: sprawdzanie zawartości katalogu

-Parametry: mkdir [katalog]

mkdir kat8 - zostanie utworzony katalog kat8

mkdir kat1 gry - zostanie utworzony katalog kat8 i gry

mkdir gry/nethack - zostanie utworzony katalog gry a w nim utworzony katalog nethack

-Opis: jest możliwość utworzenia katalogu

-Polecenia związane z plikami:

--cat -Przeznaczenie: edytowanie tekstu

-Parametry: cat [parametr] [katalog]

cat >nowy.w - wpisany tekst jest zapisany do pliku nowy.w

cat nowy.w > dwa - plik nowy.w kopiujemy do pliku dwa

-cat dwa nowy.w >tr3 - plik nowy.w i dwa (złączony jeden po drugim) kopiujemy do tr3

cat tr3 > one > two > tree - plik tr3 kopiujemy do one, one do two, two do tree

-Opis: UWAGA przy plikach binarnych grozi ta metoda zawieszeniem terminala lub systemu

--rm -Przeznaczenie: usuwanie pliku

-Parametry: rmdir [plik(i)]

rmdir plk1 - usunięcie pliku kat1

rmdir plk1 plk2 - usunięcie pliku kat1 i kat2

-Opis: jest możliwość usunięcia pliku

-Polecenia związane z kopiowaniem i przenoszeniem, plików i katalogów:

--mv -Przeznaczenie: przenoszenie plików

-Parametry: mv [plik] [cel]

mv qfile /tmp - przenosimy plik qfile do katalogu /tmp

mv f* kat - przenosimy wszystkie pliki zaczynające się literą f do katalogu kat

-Opis: za pomocą tej komendy możemy także zmienić nazwę

--cp -Przeznaczenie: kopiujemy plik

-Parametry: cp [plik/katalog] [cel]

cp plk1 plik198 - kopiujemy zawartość plk1 do plik198

cp plk1 c - kopiujemy zawartość pliku plk1 do c

-Opis: jest możliwość kopiowania plików i katalogów

--mkdir -Przeznaczenie: przenoszenie katalogu

-Parametry: mkdir [katalog] [cel]

mkdir homm3 gry - przenosimy katalog homm3 do gry

mkdir kosz /tmp - przenosimy katalog kosz do /tmp

-Opis: to polecenie działa podobnie jak mv

Polecenia związane z procesami:

-Parametry: ps [parametry]

PID TTY STAT TIME COMMAND

285 4 S 0:00 (mingetty)

287 6 S 0:00 (mingetty)

```
301 1 S 0:01 (bash)
```

399 1 R 0:00 ps

ps -aux - zotanie wyświetlone

USER PID %CPU %MEM SIZE RSS TTY STAT START TIME COMMAND

bin 95 0.0 0.6 896 44 ? S 20:32 0:00 (portmap)

```
news 296 0.0 7.9 1580 516 ? S 20:33 0:00 /sbin/innd -p5 -r
```

news 300 0.0 0.3 872 24 ? S 20:33 0:00 (overchan)

```
jasio 283 0.1 11.0 1012 716 2 S 20:33 0:01 /bin/login -- jasio
```

```
jasio 344 0.1 10.1 1220 656 2 S 20:38 0:01 -bash
```

```
asia 358 0.2 10.7 1012 700 3 S 20:39 0:01 /bin/login -- asia
```

```
asia 359 0.1 9.7 1216 632 3 S 20:39 0:00 -bash
```

```
root 1 0.3 1.6 880 104 ? S 20:30 0:04 init
```

$$[\dots]$$

```
root 465 0.0 6.2 928 404 1 R 20:53 0:00 ps -aux
```

[illegible]

USER - użytkownik procesu

PID - numer identyfikacyjny procesu

TTY - procesy związane z naszym terminalem

STAT - status procesu

R - proces działający (ang. running)

T - proces zatrzymany (ang. stopped)

P - proces w stanie spoczynku (ang. page wait)

D - proces w stanie spoczynku (ang. disk wait)

S - proces w stanie spoczynku (ang. sleeping)

I - proces w stanie spoczynku (ang. idle)

Z - proces duch (ang. zombie), zakonczył sie ale nie rozliczył
sie jeszcze z procesami nadrzednymi!

TIME - jak dlugo dziala proces

COMMAND - nazwa programu, ktorej odpowiada proces

%CPU - ilosc zaangażowania procesora w procesie

%MEM - jaka czesc pamieci operacyjnej

-Opis: jeśli chcesz zobaczyć kot jakie procesy wykonuje jest to polecenie dla ciebie

-Patrz także: [kill]

kill -Przeznaczenie: zabijanie procesów

-Parametry: kill [parametr] [PID]

kill 200 - zabija proces z numerem 200 (PID)

kill -9 200 - jeśli jest oporny i nieche się wyaczyć jest to prawie pewny sposób

kill -9 -1 - zabija wszystkie procesy procesy

kill -9 0 - !!!samobójstwo!!!

-Opis: wyłączanie procesów

-Patrz także: [ps]

Podczas analizy i wykonania ćwiczeń związanych z podstawowymi poleceniami systemu Linux, edytowaniem tekstu w trybie tekstowym oraz zarządzaniem procesami, można wyciągnąć następujące wnioski:

1. **Korzystanie z terminala:** Terminal w systemie Linux jest potężnym narzędziem do zarządzania systemem. Znajomość podstawowych poleceń i umiejętność korzystania z terminala jest niezbędna dla każdego użytkownika Linuxa.

2. **Znaki globalne:** Znaki globalne, takie jak * i ?, są przydatne do wykonywania operacji na wielu plikach naraz, co może znacząco usprawnić pracę.

3. **Podręcznik systemowy:** Korzystanie z poleceń **man** i **info** pozwala uzyskać dokładne informacje na temat dostępnych poleceń i opcji. Jest to niezwykle pomocne narzędzie, które warto wykorzystywać.

4. **Poruszanie się po systemie plików:** Znajomość specjalnych oznaczeń, takich jak **~**, **/**, **.** i **...**, ułatwia nawigację między katalogami i dostęp do różnych lokalizacji w systemie.

5. **Zarządzanie użytkownikami:** Polecenia **whoami**, **who** i **su** pozwalają na kontrolę użytkowników zalogowanych do systemu oraz przełączanie się między kontami.

6. **Zarządzanie procesami:** Polecenia **ps** i **kill** umożliwiają wyświetlanie informacji o procesach i ich zabijanie. Jest to przydatne do monitorowania i zarządzania działającymi aplikacjami.

7. **Operacje na plikach i katalogach:** Znajomość poleceń takich jak **touch**, **mv**, **cp**, **rm**, **mkdir**, **rmdir**, **cat**, **ls**, **pwd** pozwala na efektywne zarządzanie plikami i katalogami w systemie.

8. **Edytory tekstu w trybie tekstowym:** Edytory takie jak **nano** i **vi** są przydatne do edycji plików tekstowych z poziomu konsoli. Znajomość podstawowych poleceń tych edytorów jest przydatna.

9. **Strumienie i potoki:** Znajomość znaków specjalnych, takich jak **|**, **<**, **>**, **>>**, pozwala na przekierowywanie strumieni danych i wykonywanie potoków poleceń, co jest bardzo przydatne w bardziej zaawansowanych zadaniach.

10. **Wyszukiwanie plików:** Polecenie **find** oraz **locate** ułatwiają wyszukiwanie plików w systemie na podstawie różnych kryteriów, co jest przydatne do organizacji i zarządzania plikami.

11. **Edycja i przeglądanie plików tekstowych:** Umiejętność pracy z plikami tekstowymi w trybie tekstowym za pomocą edytorów jak **nano**, **vi**, **grep**, **cat** jest kluczowa w środowisku Linux.

12. **Zarządzanie procesami:** Zrozumienie procesów, ich identyfikatorów (PID) i umiejętność zarządzania nimi za pomocą poleceń **ps** i **kill** jest ważne dla utrzymania stabilności i wydajności systemu.

13. **Narzędzia do zarządzania systemem:** Linux oferuje wiele narzędzi i poleceń do zarządzania systemem plików, użytkownikami, procesami i innymi aspektami systemu.

Wnioski ogólne: Zrozumienie i praktyczna umiejętność korzystania z podstawowych poleceń systemu Linux oraz narzędzi do zarządzania systemem jest kluczowe dla skutecznego użytkowania tego systemu operacyjnego. Praca z terminalu może być potężnym narzędziem dla zaawansowanych

użytkowników i administratorów systemów. Regularna praktyka i eksperymentowanie z różnymi poleceniami pomogą w doskonaleniu tych umiejętności.