

System operacyjny UNIX

Andrzej Bobyk

Spis treści

1	Literatura	1
2	Wprowadzenie	2
2.1	Charakterystyka systemu UNIX	2
2.2	Otwieranie i zamykanie sesji	2
3	System plików	4
4	Polecenia systemu	5
4.1	Wzorce i znaki specjalne	7
4.2	Strumienie i potoki	7
5	Edytor vi	7
5.1	Polecenia edytora vi	8
6	Kontrola zadań, skrypty, pliki konfiguracyjne	11
7	Komunikacja z innymi komputerami	11

1 Literatura

- Peter P. Silvester: *System operacyjny UNIX*, WNT, Warszawa 1990
- Zbigniew Kacprzyk (red.): *Łagodne wprowadzenie do systemu UNIX*, Politechnika Warszawska, Warszawa 1991
- Douglas Topham, UNIX, Intersoftland/Prentice Hall Int., Warszawa/Hemel Hempstead 1995
- Stephen Prata, Donald Martin, *Biblia systemu UNIX V. Polecenia i programy użytkowe*, LT&P, Warszawa 1994
- Alan Southerton, Edwin C. Perkins, Jr., *Słownik poleceń systemów UNIX i X*, WNT, Warszawa 1995

2 Wprowadzenie

2.1 Charakterystyka systemu UNIX

- UNIX jest systemem **wielodostępnym** przeznaczonym dla wielu użytkowników (ang. *users*).
- UNIX jest systemem **wielozadaniowym**, tzn. jednocześnie może być wykonywanych wiele programów, należących do różnych użytkowników.
- Każdy użytkownik posiada swój unikalny **identyfikator użytkownika** (ang. *user id, login name*) przydzielony mu przez **administratora** (ang. *root*) oraz **hasło** dostępu (ang. *password*), które może (i powinien w odstępach co minimum 6–8 tygodni) zmieniać. Zasady wyboru haseł i ich ochrony wiążą się ściśle z ochroną systemu jako całości. Nie należy więc hasła udostępniać osobom trzecim ani też zapisywać go, powinno ono mieć od 6 do 8 znaków i zawierać co najmniej jeden znak specjalny (nie-literę, tzn. cyfrę, znak interpunkcyjny itp.). Powinno ono także być trudne do odgadnięcia – w szczególności nie wolno wybierać jako hasła swojego identyfikatora (nawet nieznacznie zmienionego, np. poprzez dostawienie cyfry lub wpisywanego od końca), swojego nazwiska lub imienia (ew. współmałżonka), daty urodzenia, numeru telefonu itp. Należy również pamiętać o zakończeniu sesji przy odejściu od końcówki (konsoli lub terminala).
- System UNIX rozróżnia małe i duże litery!
- Każdy użytkownik należy do określonej **grupy** (ang. *group*).
- Komputer UNIX-owy jest zazwyczaj włączony w sieć, umożliwiając tym samym pracę zdalną.
- Systemy UNIX-owe posiadają **graficzny interfejs użytkownika**, zwany X-Windows. Można z niego korzystać przy **konsoli** (ang. *console*) lub poprzez odpowiednio wyposażony terminal (X-Terminal).
- Komputer UNIX-owy nie może być wyłączany bez uprzedniego zamknięcia systemu przez administratora ze względu na możliwość utraty danych i uszkodzenia systemu jako całości!

2.2 Otwieranie i zamykanie sesji

Pracę z systemem rozpoczynamy od **otwarcia sesji** (ang. *logging in*):

- W odpowiedzi na **zachętę (monit)** (ang. *prompt*)

login:

należy podać swój identyfikator¹.

- Następnie system zażąda podania hasła, wyświetlając na ekranie monit

Password:

- Znaki hasła nie ukazują się na ekranie, aby osoby postronne nie poznały go.
- W przypadku popełnienia błędu podczas wprowadzania identyfikatora bądź hasła system wyświetli odpowiedni komunikat – należy wówczas ponowić próbę otwarcia sesji.
- Po otwarciu sesji system wyświetla linię zachęty, np.

```
fizpc84:~>
```

i jest gotów do przyjmowania poleceń. Polecenia są interpretowane poprzez tzw. **powłokę** (ang. *shell*), stanowiącą pomost między użytkownikiem a **jądrem** systemu (ang. *kernel*). Istnieją różne powłoki, np.:

- **sh** – Bourne shell, oraz jego derywacje: **ksh** – Korn shell i **bash** – Bourne again shell, wzbogacone o ułatwienia do pracy interakcyjnej.
- **cs**h – C shell i jego rozszerzenie **tcsh** - The C shell.

tcsh, **bash** i **ksh** udostępniają **historię** poleceń i umożliwiają posługiwanie się nią przy pomocy klawiszy kursora.

- Przy korzystaniu z systemu X-Windows na **pulpicie** angdesktop ukażą się m.in. rozwijalne menu, okna, ikony itd. Zachęta ukaże się wówczas w oknie, zwanym **oknem terminala**. Jeżeli takowe nie zostanie otwarte należy to uczynić wybierając pozycję z menu narzędziowego.
- Sesję zamykamy wydając komendę **exit** lub **logout**. W przypadku pracy okienkowej konieczne jest ponadto zamknięcie sesji X-Windows.
- Niektóre opcje menu są dostępne przez kliknięcie prawym klawiszem myszy na pulpicie.

3 System plików

- System plików zawiera²:

– **pliki zwykłe** (ang. *regular files*) [-],

¹Przy otwieraniu sesji na konsoli można też dwukrotnie kliknąć lewym klawiszem myszy na ikonce ze swoim identyfikatorem

²Patrz opis polecenia **ls**

- **katalogi** (ang. *directories*) [d],
 - **powiązania symboliczne** (ang. *symbolic links*) [l],
 - **pliki specjalne** (ang. *special files*), np. dla urządzeń [b, c, p].
- Każdy plik posiada swojego **właściciela** (ang. *owner*).
 - Sumaryczna liczba i wielkość plików należących do jednego użytkownika może być ograniczona poprzez **limit** (ang. *quota*).
 - Właściciel może określać **prawa dostępu** (ang. *access rights*) do swoich plików dla innych użytkowników lub grup.
 - Pliki mogą mieć długie nazwy, w szczególności zawierające jedną lub więcej kropek i zaczynające się od kropki.
 - Katalogi mają strukturę **drzewa**, rozpoczynającą się od **korzenia** (ang. *root directory*), oznaczonego symbolem „/” (ang. *slash*). Katalogi są **węzłami** drzewa, zaś pliki jego **liśćmi**.
 - Każdy użytkownik posiada swój **katalog macierzysty (domowy)** (ang. *home directory*), w którym rozpoczyna pracę.
 - Nazwa „.” określa bieżący katalog, „..” katalog znajdujący się o jeden stopień wyżej w hierarchii (ang. *parent directory*), „~” – katalog macierzysty użytkownika, zaś „~jasio” katalog macierzysty użytkownika *jasio*.
 - Każdy plik posiada swoją **nazwę absolutną**, określoną przez **ścieżkę absolutną** (ang. *absolute (full) path*) będącą sekwencją nazw katalogów rozpoczynającą się od „/” i oddzielonych tymże znakiem, jak również nazwę względną, zapisaną przy pomocy symboli „.”, „..” oraz „~”.
 - Prawa dostępu do plików są określone trójpoziomowo: dla właściciela pliku, jego grupy oraz pozostałych użytkowników. Istnieją trzy rodzaje praw dostępu²:
 - **czytania** (ang. *read*) [r],
 - **pisania** (ang. *write*) [w],
 - **wykonywania** (ang. *execute*) [x] programu ew. przeglądania katalogu.

4 Polecenia systemu

Składnia poleceń generalnie jest następująca:

nazwa -opcje argumenty

- Składniki polecenia (nazwę, zestaw opcji i argumenty) oddziela się od siebie **znakami odstępu (spacjami)**.
- Opcje mogą być **grupowane**.

Najważniejsze polecenia to:

- `man polecenie` – wyświetla stronę podręcznika (ang. *manual, man page*) poświęconą *poleceniu*. W szczególności `man man`. `man -k klucz` (lub też *apropos klucz* wyświetla listę stron podręcznika, na których występuje *klucz*).
- `cd katalog` – przejście do *katalogu* (ang. *change directory*).
- `mkdir katalog` – utworzenie nowego *katalogu* (ang. *make directory*).
- `rmdir katalog` – usunięcie pustego *katalogu* (ang. *remove directory*).
- `ls` – wyświetlenie zawartości bieżącego katalogu (ang. *list*). Podanie opcji `-la` powoduje wyświetlenie wszystkich nazw plików w danym katalogu (również tych zaczynających się od kropki) w długim formacie, np.:

```
-rw-r--r--    1 bobyk    user           26 Jan 15 15:30 .forward
-rw-----    1 bobyk    user          2719 Feb 20 22:59 .history
drwxr-xr-x    2 bobyk    user           28 Jan 14 12:46 .icons
-rw-r--r--    1 bobyk    user           862 Jan 15 15:32 .login
drwx-----    3 bobyk    user          4096 Feb 19 15:55 .netscape
-rw-r--r--    1 bobyk    user            40 Jan 13 17:59 .rhosts
drwxr-xr-x    2 bobyk    user           142 Jan 13 14:41 Desktop
-rw-r--r--    1 bobyk    user            69 Jan 10 13:44 bobyk.outbox
-rw-r--r--    1 bobyk    user        111708 Feb 20 20:45 core
drwx-----    8 bobyk    user           4096 Feb 19 16:02 pine3.95
-rw-----    1 bobyk    user       2154473 Feb 19 16:01 pine3.95.tar.gz
drwx-----    2 bobyk    user            9 Feb 20 19:54 przyklad
drwxr-xr-x    4 bobyk    user           105 Jan 14 13:55 public.html
-rwx-----    1 bobyk    user          27996 Feb 17 18:47 winmine.exe
```

Pierwsza kolumna określa typ pliku i prawa dostępu, następne właściciela, grupę, wielkość w kilobajtach, datę ostatniej modyfikacji i nazwę.

- `mv stare nowe` – zmiana nazwy pliku/katalogu ze *stare* na *nowe*, ew. przeniesienie pliku/katalogu *stare* do istniejącego katalogu *nowe* (ang. *move*).
- `cp stare nowe` – kopiowanie pliku/katalogu, jak powyżej (ang. *copy*).
- `rm plik` – usunięcie *pliku* (ang. *remove*).

- `lp plik` – wysłanie *pliku* na drukarkę.
- `pwd` – wyświetlenie nazwy bieżącego katalogu (ang. *print working directory*).
- `passwd` – zmiana hasła (ang. *password*).
- `chmod prawa plik` – zmiana *praw* dostępu do pliku (ang. *change mode*).
- `chown właściciel plik` – zmiana *właściciela* pliku (ang. *change owner*).
- `find katalog kryterium` – wyszukuje pliki w *katalogu* (wraz z podkatalogami) na podstawie *kryterium*.
- `cat plik` – wyświetla zawartość *pliku* (tekstowego) na ekranie (ang. *concatenate*).
- `more plik` – j.w., ale ze stronicowaniem.
- `grep wzorzec plik` – wyświetla linie w *pliku*, zawierające dany wzorzec.
- `vi plik` – edycja *pliku* (ang. *visual editor*).
- `who` – wyświetla nazwy użytkowników, pracujących w systemie.
- `finger @komputer` – j.w., ale na *komputerze* zdalnym.
- `telnet komputer` – rozpoczęcie pracy na *komputerze* zdalnym.
- `ftp komputer` – transfer plików z/do *komputera* zdalnego (ang. *file transfer protocol*).
- `set zmienna{=wartość}` – ustawienie *zmiennnej*, z ewentualnym nadaniem jej *wartości*.
- `setenv zmienna {wartość}` – j.w., ale dla *zmiennnej* środowiska (ang. *environment*).
- `alias skrót polecenie` – utworzenie *skrót*u często używanego *polecenia*.

4.1 Wzorce i znaki specjalne

Przy podawaniu nazw plików można stosować **wzorce** (ang. *wildcards*):

- `?` – reprezentuje dowolny pojedynczy znak,
- `*` – reprezentuje dowolną liczbę (w szczególności zero) znaków,
- `[lista]` – reprezentuje dowolny znak z *listy*.
- `{lista}` – reprezentuje dowolny z łańcuchów *listy*.

Niektóre znaki mają dla powłoki specjalne znaczenie. Są to, oprócz wzorców, znaki: `*$*[]?{}~_<>&!|;()\`'“` oraz znaki odstępu (spacji), tabulacji i nowej linii. Ich użycie w sposób dosłowny wymaga poprzedzenia każdego z nich znakiem „\” (ang. *backslash*). Umieszczenie łańcucha znaków w pojedynczych ' lub podwójnych cudzysłowach " powoduje, że tylko niektóre z tych znaków są interpretowane przez powłokę.

4.2 Strumienie i potoki

UNIX umożliwia zmianę standardowego wejścia (ang. *input*) (klawiatura) oraz standardowego wyjścia (ang. *output*) (ekran) programu z/do innego urządzenia lub pliku, jak również przekierowanie wyjścia jednego programu na wejście drugiego poprzez tzw. **potoki** (ang. *pipeline*):

- `program < plik` – *plik* staje się źródłem informacji wejściowych do *programu*,
- `program > plik` – wynik działania *programu* zostanie zapisany w *pliku*,
- `program » plik` – j.w., ale z żądaniem **dopisania** do już istniejącego *pliku*,
- `program1 | program2` – przekierowanie wyjścia *programu1* na wejście *programu2*.

5 Edytor vi

Edytor *vi* jest *standardowym* programem systemu UNIX umożliwiającym pracę pełnoekranową i niezastąpionym przy korzystaniu ze zdalnego terminala o ograniczonych możliwościach lub powolnym dostępie. Edytor *vi* ma dwa tryby pracy: poleceń (komend), w którym edytor znajduje się po uruchomieniu (`vi plik`) i tekstowy (wstawiania). W trybie poleceń wprowadzane znaki nie pojawiają się na ekranie (z wyjątkiem znaków `!:/?`) i są interpretowane jako operacje na tekście (kasowanie znaków, linii itp.). W trybie tekstowym, oprócz wprowadzania znaków możliwe są następujące polecenia:

POLECENIE	DZIAŁANIE
<code><ESC></code>	powrót do trybu komend
<code>[←], <CTRL->H</code>	skasowanie znaku ³
<code><CTRL->W</code>	skasowanie słowa
<code><CTRL->V</code>	potraktowanie następnego znaku literalnie
<code><ENTER></code>	przejsie do nowej linii

³faktyczne skasowanie znaków z ekranu następuje dopiero w momencie ich nadpisania lub powrotu do trybu komend

5.1 Polecenia edytora vi






- Polecenia powodujące przejście do trybu wstawiania:

POLECENIE	DZIAŁANIE
i	wpisywanie tekstu przed kursorem
I	wpisywanie tekstu przed początkiem linii
a	wpisywanie tekstu za kursorem
A	wpisywanie tekstu za końcem linii
o	otworzenie nowej linii poniżej kursora
O	otworzenie nowej linii powyżej kursora
R	zastępowanie tekstu (nadpisywanie)
s	zastąpienie nowym tekstem jednego znaku na pozycji kursora
S	zastąpienie nowym tekstem całej bieżącej linii
<i>czakres</i>	zmiana wskazanego tekstu
C	zastąpienie tekstu od pozycji kursora do końca linii

- Polecenia operacji na tekście bez przejścia do trybu wstawiania:

POLECENIE	DZIAŁANIE
<i>dzakres</i>	skasowanie wskazanego tekstu
dd	skasowanie bieżącej linii
x	skasowanie znaku na pozycji kursora
X	skasowanie znaku przed kursorem
<i>yzakres</i>	skopiowanie tekstu do bufora pomocniczego
Y	skopiowanie bieżącej linii do bufora pomocniczego
p	wstawienie tekstu z bufora za kursorem lub poniżej niego
P	wstawienie tekstu z bufora przed kursorem lub powyżej niego
J	połączenie linii bieżącej z następną
r	zastąpienie znaku na pozycji kursora jednym znakiem
u	anulowanie ostatniej dokonanej zmiany tekstu
.	powtórzenie ostatniego polecenia zmieniającego tekst

- Polecenia poruszania się po tekście i wyszukiwania:

POLECENIE	DZIAŁANIE
O	przeniesienie kursora na początek linii
\$	przeniesienie kursora na koniec linii
h , <CTRL-> H ,  , 	przeniesienie kursora o jeden znak w lewo
j , <ENTER>, 	przeniesienie kursora o jedną linię w dół
k , 	przeniesienie kursora o jedną linię w górę
l , <SPACJA>, 	przeniesienie kursora o jeden znak w prawo
<CTRL-> F	przesunięcie okna o jeden ekran w dół
<CTRL-> B	przesunięcie okna o jeden ekran w górę
<CTRL-> D	przesunięcie okna o pół ekranu w dół
<CTRL-> U	przesunięcie okna o pół ekranu w górę
G	przeniesienie kursora do ostatniej linii w pliku
nG	przeniesienie kursora do <i>n</i> -tej linii w pliku
<i>/wzorzec</i>	szukanie <i>wzorca</i> w przód
<i>?wzorzec</i>	szukanie <i>wzorca</i> w tył

- Polecenia trybu liniowego (edytora `ex`):

POLECENIE	DZIAŁANIE
:q	zakończenie działania edytora
:q!	zakończenie działania edytora z pominięciem zmian
:wq , :x , ZZ	zakończenie działania edytora z zapisaniem zmian
:w nazwa	zapisanie pliku pod podaną <i>nazwą</i>
:w! nazwa	zapisanie pliku pod podaną <i>nazwą</i> już istniejącą
:s/stare/nowe/g	globalne zastąpienie wzorca <i>starego</i> <i>nowym</i>
:set opcja {wartość}	ustawienie <i>opcji</i> z ew. nadaniem jej <i>wartości</i>
:set showmode	sygnalizacja trybu pracy edytora
:set number	włączenie numeracji linii
:unset opcja , set noopcja	wyłączenie opcji
:shell	wywołanie powłoki

6 Kontrola zadań, skrypty, pliki konfiguracyjne

Bieżące polecenie może być zarówno przerwane kombinacją klawiszy <CTRL->C lub zawieszona (<CTRL->Z). Zawieszona polecenie może być kontynuowane (komenda `fg`) lub można zlecić jego dalsze wykonanie w tle (komenda `bg`). Możliwe jest również bezpośrednie zlecenie powłoce wykonania polecenia w tle poprzez umieszczenie znaku `&` (ang. *ampersand*) po poleceniu. Komenda `jobs` wyświetla listę zawieszonych lub wykonujących się w tle poleceń.

Zestawy poleceń mogą być łączone w *skrypty*, będące w istocie programami, napisanymi w języku powłoki. Szczególnymi skryptami, wykonywanymi przy rozpoczęciu sesji są pliki `.login` oraz `.cshrc`, umieszczone w katalogu macierzystym użytkownika. Służą one do

indywidualnego dopasowania własnego środowiska pracy. Różnią się one tym, że skrypt `.login` jest wykonywany tylko raz przy otwarciu sesji, natomiast `.cshrc` każdorazowo przy rozpoczęciu pracy interpretatora poleceń (powłoki) `cs` bądź `tcsh`. Odpowiednikiem tych dwóch skryptów dla powłoki `sh` i pochodnych jest plik `.profile`.

Innymi przykładami plików konfiguracyjnych są `.exrc`, zawierający komendy edytora `vi` i wykonywany przy jego uruchomieniu, `.xinitrc` oraz `.Xdefaults` i `.Xresources`, konfigurujące środowisko pracy X-Windows, `.mailrc` konfigurujące pocztę itd.

7 Komunikacja z innymi komputerami

Połączenie komputerów w sieć umożliwia zarówno pracę zdalną na innej maszynie, transfer plików itd. Do otworzenia sesji na komputerze zdalnym służy komenda `telnet adres`, gdzie *adres* jest adresem sieciowym komputera zdalnego. Następnie należy w sposób opisany w punkcie 2.2 otworzyć sesję zdalną. Zamknięcie sesji zdalnej powoduje jednocześnie zakończenie działania programu `telnet`. Jeżeli na komputerze lokalnym i zdalnym identyfikatory użytkownika są identyczne można również użyć polecenia `rlogin adres`. Podobnie rozpoczynamy transfer plików z/do innego komputera poleceniem `ftp adres`. Monit systemu zmienia się na `ftp>` i można wprowadzać następujące polecenia:

- `cd katalog`, `mkdir katalog`, `rmdir katalog` – zmiana, utworzenie, skasowanie *katalogu* na komputerze zdalnym.
- `lcd katalog` – zmiana *katalogu* na komputerze lokalnym.
- `dir katalog` – wyświetlenie zawartości *katalogu* zdalnego.
- `!polecenie` – wykonanie *polecenia* lokalnie.
- `get (put) plik` – przetransferowanie *pliku* z komputera zdalnego na lokalny lub vice versa.
- `mget (mput) lista plików` – przetransferowanie kilku *plików* jednocześnie z komputera zdalnego na lokalny lub vice versa.
- `ascii` – przełączenie na 7-bitowy tryb transferu ASCII.
- `bin` – przełączenie na 8-bitowy (binarny) tryb transferu.
- `?`, `help` – pomoc.
- `bye`, `quit` – zakończenie sesji FTP.