

## 4. Wprowadzenie do Systemów Operacyjnych UNIX i LINUX

### 4.1. Historia Unixa i Linuxa, instalacja systemu.

Historia UNIX-a rozpoczyna się w połowie lat sześćdziesiątych XX wieku od MULTICSa - wielkiego systemu operacyjnego z podziałem czasu, opracowanego w MIT (Massachusetts Institute of Technology) we współpracy z firmami amerykańskimi General Electric oraz Bell Laboratories, dla komputera GE 645 [1]. MULTIX został napisany prawie w całości w języku PL/1 (300 000 wierszy kodu źródłowego) i rozszerzony do środowiska wieloprocesorowego. Przedsięwzięcie było ważne dla późniejszego rozwoju systemów operacyjnych, w szczególności dla systemu UNIX, lecz samo nigdy nie zostało uwieńczone pełnym sukcesem; prace zarzucono w 1969.

Massachusetts Institute of Technology (MIT), jest znaną politechniką amerykańską o dużych zasługach dla rozwoju informatyki. Tam powstały takie wynalazki i opracowania, jak tranzystor, układ scalony, język Lisp, język Clu opracowany w celu wyrażania abstrakcji danych, język Argus, systemy CTSS i MULTICS, szyfr RSA, protokół uwierzytelniania Kerberos, system X-Window, przeglądarka WWW. MIT powstał w 1861, został założony przez Williama Bartona Rogersa, geologa i utalentowanego nauczyciela.

System operacyjny UNIX został opracowany przez Kena Thompsona z AT&T Bell Laboratories . Jest przeznaczony głównie dla komputerów pełniących rolę serwerów. Pomimo że powstał pod koniec lat 60. (1969), nadal jest jednym z częściej spotykanych w świecie systemów operacyjnych. Większość producentów superkomputerów dołącza do nich własną wersję tego systemu. Na bazie UNIX-a powstał też Linux, czyli system przeznaczony dla komputerów osobistych, choć jest on również z powodzeniem używany w komputerach świadczących różne usługi sieciowe.

W roku 1970 w/w Ken Thompson oraz D. Ritche przepisali UNIX w C czyniąc go przenośnym . Początkowo był instalowany nieodpłatnie na komputerach w szkołach wyższych. W roku 1980 firma Microsoft opracowała wersję UNIX nazwaną XENIX i przeznaczoną do instalacji na komputerach osobistych (PC). Około roku 1985 zdefiniowano standard UNIX-a.

Aktualne najbardziej znane dystrybucje UNIX-a to:

BSD (Berkeley Software Distribution) przeznaczona dla komputerów Sun (SUNOS),

AIX dla stacji roboczych IBM oraz

XENIX dla komputerów osobistych. Ta ostatnia wersja została obecnie prawie całkowicie wyparta przez rozwijający się z niezwykłą dynamiką LINUX.

**LINUX** jest systemem operacyjnym napisanym na bazie UNIXa przez Linusa Torvaldsa z Uniwersytetu w Helsinkach w 1991 roku. W systemie Linux można pracować zarówno w interfejsie tekstowym jak i graficznym (X-Windows). System jest rozpowszechniany darmowo, stąd jego coraz większa popularność. Z sieci Internet można ściągnąć kod źródłowy systemu. Linux jest alternatywą dla systemów operacyjnych firmy Microsoft. Oprócz tego, że jest darmowy, jest także uznawany za system stabilniejszy, o wiele lepiej zabezpieczony przed włamaniami. Jego wadą natomiast jest nieco trudniejsza konfiguracja i użytkowanie, co powoduje że Linux jest uważany (w opinii wielu osób niesłusznie) za system przeznaczony dla bardziej zaawansowanych użytkowników.

Obecnie liczne wersje Linuksa są na ogół zgodne ze standardem **POSIX** (Portable Operating System Interface for Computer Enviroments). Posiadają dzięki temu zdolność pracy z różnymi komputerami w sieci. Niezależność programów usługowych i użytkowych od typu komputera i procesora zapewnia otwarty dostęp do wersji źródłowej systemu (ang. OPEN SOURCE.)

Istnieje wiele wersji (tzw. dystrybucji) Linuksa. Do najbardziej znanych należą: Mandrake, Debian, Redhat, Suse, Open Linux Lite firmy CALDERA, Aurox.. Większość tych wersji można zdobyć bezpłatnie poprzez Internet, zaglądając na przykład pod adresy: [www.redhat.com](http://www.redhat.com) , [www.linux.org.pl](http://www.linux.org.pl) – (klubu miłośników Linuksa w Polsce), [www.suse.de](http://www.suse.de), [www.sunsite.com](http://www.sunsite.com) i wiele innych. W chwili oddania tego skryptu do druku ostatnią wersję Auroxa 11 można znaleźć w numerze 57 z 2005 roku czasopisma Linux+ Extra.

### **Instalacja Auroksa 9.4 - uwagi ogólne**

Auroks może być zainstalowany jako kolejny system operacyjny - nie musimy rezygnować na przykład z systemu Windows, który jest często zainstalowany na naszym komputerze. Możemy również zachować wszystkie nasze dane i programy, które wykorzystywaliśmy pracując z naszym pierwotnym systemem operacyjnym. Instalacja może być przeprowadzona w trybie graficznym lub w trybie tekstowym. **Cała procedura instalacyjna jest opisana szczegółowo i przystępnie w [4].**

Przed zainstalowaniem systemu musimy zwolnić odpowiednią ilość miejsca na dysku - potrzeba co najmniej 600 MB pamięci na sam system i dodatkowo tyle MB pamięci, ile chcemy przeznaczyć na nasze dane i programy. W praktyce potrzeba, jak sędzę, co najmniej 1 GB pamięci na dysku. Inne wymagania systemowe dotyczą procesora - do pracy w trybie graficznym potrzebny jest procesor 400 MHz Pentium II lub lepszy oraz minimum 128 MB pamięci RAM, zalecane 256 MB lub więcej. Do pracy w trybie tekstowym wymagania są nieco skromniejsze - wystarczy procesor zgodny z Pentium, 200 MHz lub lepszy i 64 MB pamięci RAM.

## 4.2. Opis Systemów Operacyjnych UNIX i LINUX

Zestaw procedur składających się na Unix (i Linux) – dzieli się zasadniczo na 4 główne części:

A. Jądro (ang. Kernel) jest głównym składnikiem systemu. Jądro uruchamia programy, przydziela pamięć i inne zasoby procesom, zarządza obszarem wymiany, steruje dyskami, komunikacją z siecią, z urządzeniami zewnętrznymi itd. Stanowi odpowiednik jądra w DOS –ie. Wraz z niewielką grupą programów systemowych takich jak pliki konfiguracyjne, inne pliki ukryte rozpoczynające się na . tworzy to co zwykle się określa nazwą GNU/Linux.

Powłoka - zawiera oprogramowanie tzw. graficznego interfejsu użytkownika (ang. GUI – Graphics User Interface.)

Powłoka służy do interpretowania poleceń i przekazywania ich od użytkownika do jądra.

Wykorzystuje się do tego celu wiersz poleceń lub interfejs graficzny będący odpowiednikiem systemu X – Windows w UNIX-e lub systemu „okienek” w „Microsoft Windows”

B. System plików systemowych - obejmuje większość programów niezbędnych do administrowania systemem operacyjnym, kontroli porządku rozmieszczenia plików na dysku itp., na przykład cp (kopiuj) , mv (przenieś), df (pokaż wielkość miejsca na dysku).

Drzewiasty system tworzy uporządkowaną strukturę plików:

- 1) zwykłych
- 2) katalogów
- 3) plików specjalnych

**Katalogi** standardowo zarezerwowane są dla plików systemu i własnych plików użytkownika, same katalogi są też plikami zawierającymi spisy plików

**Pliki Specjalne** zarządzają urządzeniami WE / WY, strumieniami danych, definiują znaki itp.

C. Programy użytkowe (aplikacje) są to programy wykorzystywane przez użytkowników do pracy z komputerem: edytory (vi, emacs), kompilatory, gry, programy komunikacyjne, naukowe itp.

### Niektóre zalety Linuksa (Unixa):

- 1) Język poleceń jest prosty, osoby znające język angielski z łatwością identyfikują kilkuliterowe skróty wyrazów tego języka
- 2) Duży wybór narzędzi programowych dla środowiska systemowego dostępnego w wersji źródłowej
- 3) Niezależność programów usługowych i użytkowych od typu komputera i procesora - fakt, że są to programy źródłowe (Open Source) umożliwia ich kompilację i

uruchamianie praktycznie na każdym komputerze. Oczywiście poprawianie kodu źródłowego wymaga pewnej znajomości tego kodu – nie zawsze wystarczy zwykła modyfikacja instrukcji wskazanych przez debugger kompilatora .

- 4) System jest w zasadzie darmowy, kupujemy go w cenie nośników pamięci, jako dodatek do popularnych czasopism lub ściągamy z Internetu.
- 5) System zapewnia wysoki poziom bezpieczeństwa, włamania są utrudnione, system operacyjny jest dosyć odporny na błędy użytkowników)

Generalnie można stwierdzić, że używanie Linuksa wymaga większego przygotowania teoretycznego i większego doświadczenia. Ocenia się, że Linuks jest wykorzystywany w kilku procentach na komputerach osobistych i w około 30 procentach na serwerach. W Polsce system ten jest nieco mniej popularny. Trudniej jest więc rozwiązywać ewentualne problemy – choć istnieje na rynku wiele doskonałych książek dotyczących Linuksa a poprzez Internet można znaleźć odpowiedź na prawie każde pytanie.

Zasadniczo wyróżnić można **dwa tryby pracy** z systemem operacyjnym Linux:

- Tryb tekstowy (poprzez interfejs tekstowy, ang. Character cell interface)
- poprzez interfejs graficzny.

Wybrane cechy wersji Linuksa o nazwie RedHAT 7.3

- Prosta instalacja systemu operacyjnego oraz plików użytkowych i systemowych typu **\*.rpm** – jednym rozkazem **rpm nazwa.rpm**, który rozpakowuje i instaluje podprogramy składowe pliku **nazwa.rpm** w odpowiednich katalogach, istniejących lub tworzonych w trakcie instalacji
- Zawiera program Open Office będący odpowiednikiem popularnego pakietu programów biurowych MS Office firmy Microsoft, a także dużą ilość innych, darmowych programów graficznych, multimedialnych, komunikacyjnych, np. Kudzu – program do wykrywania i konfigurowania dołączanych urządzeń (hardware) itp.
- Nieduże wymagania sprzętowe w wielu przypadkach wystarczy procesor Intel 386, Pentium lub młodszy, 16 MB RAM, 500 MB pamięci na dysku oraz stacja CD ROM. Wymagania te w przypadku Auroksa 11 są nieco większe i zostały podane w podrozdziale 4.1. przy omawianiu instalacji.

### 4.3. Niektóre polecenia Linuksa

W Linuksie polecenia są plikami wykonywalnymi.

`cd nazwa_katalogu` - zmień katalog bieżący na podkatalog o nazwie `nazwa_katalogu`

`cd..` – idź do katalogu „piętro wyżej”

`cp nazwa1 ~/katalogx/nazwa2` - kopiuje plik `nazwa1` do katalogu `katalogx` nadając mu nazwę `nazwa2` ; w niektórych wersjach linuxa trzeba podać zamiast `~` pełną nazwę ścieżki prowadzącej od katalogu tzw. korzenia / (ang. root) do katalogu o nazwie `katalogx`. Katalog korzenia / znajduje się na szczycie nadrzędno – podrzędnej, hierarchicznej struktury katalogów zawierających różne pliki.

Nazwa katalogu `root` może nastręczyć początkowemu użytkownikowi linuxa pewne kłopoty wynikające z faktu, że ta sama nazwa (`root`) jest standardowo przydzielana przez system administratorowi (ang. superuser). Oznacza nazwę administratora jako „specjalnego” użytkownika. Na szczęście kontekst występowania obydwu nazw `root` jest zazwyczaj różny i po pewnym czasie pracy z linuxem nie zachodzą już nieporozumienia.

`mv nazwa1 nazwa2` zmień nazwę pliku

`ls -l` listuj (wyświetl nazwy plików) znajdujących się w bieżącym katalogu w postaci „l” ( skrót od ang. long) czyli długiej, pełnej, podając je w porządku alfabetycznym i specyfikując prawo dostępu, nazwę właściciela pliku, nazwę grupy, rozmiar pliku w bajtach, datę i czas ostatniej modyfikacji oraz nazwę pliku (`moj_plik`).

Po komendzie `ls -l` na ekranie zobaczymy na przykład:

```
-rw-rw-r-- 1 wsosn zmit 32000 july 7 8:43 moj_plik
```

co oznacza, że w bieżącym katalogu jest tylko jeden plik o nazwie `mój_plik`.

`ls -ltr` wyświetla te same informacje ale w innej kolejności – w tym przypadku kierując się czasem ostatniej modyfikacji pliku (`t` – ang. time) oraz stosując kolejność „od najstarszej do najmłodszej modyfikacji” (`r` – ang. reverse)

`alias rozkaz_nasz = 'rozkaz_linuxa'` zastępuje `rozkaz` w apostrofach innym, zdefiniowanym samodzielnie, na przykład

```
alias ll = 'ls -l'
```

`alias bb = 'telnet 145.83.96.100'` – jak wyżej, wygodna forma pozwalająca szybko połączyć się z wybranym komputerem

`unalias ll` odwołanie w/w rozkazu `alias ll`

`mkdir nazwa_katal` utwórz katalog o nazwie `nazwa_katal`

`chmod plik1 +x` nadaj zbiorowi o nazwie `plik1` prawo wykonywania

`rm *.*` usuń wszystkie zbiory z bieżącego katalogu (to znaczy z tego katalogu, w którym jesteś w danej chwili)

`rmdir kat1` usuń katalog o nazwie `kat1`

`rm -i *.*` (`-i = -inquire`) usuń wszystkie zbiory z bieżącego katalogu upewniając się przy każdym zbiorze, czy rzeczywiście chcesz go usunąć. Dla bezpieczeństwa warto

zdefiniować alias: `alias rm = 'rm -i'` który pozwoli uniknąć przypadkowego skasowania potrzebnego zbioru.

`rm -rf` usuń cały katalog razem z zawartością

`cat nazwa 1 nazwa 2 > nazwa 3` połącz zbiory

`startx` rozkaz uruchamiania interfejsu graficznego (powłoki)

`su „super user”` zalogowanie się jako administrator. Zachodzi potrzeba podania hasła administratora.

`pwd` Gdzie ja jestem, jaka jest nazwa katalogu w którym obecnie znajduję się?

`cd` - powrót do home directory z dowolnego miejsca

Można adresować wyniki poleceń: na ekran, drukarkę, do innego pliku

`ls > nazwa_pliku_zawierającego_spis_plików`

`ls -al | less` - pokazuj pliki stronami

`ls -al | lpr` - wyślij listę na drukarkę

`ls -al /mnt/cdrom/READ*` - wyślij listę zbiorów zaczynających się na READ i znajdujących się na dysku kompaktowym.

### 4.3. Uruchamianie LINUXSA

Procedura startowa jest podobna do procedury startowej systemu WINDOWS.

(Power on Self Test – POST itd.) W wersji Linuxa Redhat wykorzystywany jest standardowo program o nazwie LILO (skrót od Linux Loader), umożliwiający wybór dysku startowego, systemu operacyjnego itp. Jest też możliwość wykorzystania w tym celu innych programów.

Jak już wspomnieliśmy możemy wykorzystywać interfejs graficzny np. GNOME, uruchamiany rozkazem **startx** lub pracować w trybie tekstowym.

Praca pod GUI wymaga umiejętności powszechnie znanych wszystkim użytkownikom systemu operacyjnego WINDOWS:

- praca z myszą: wskaż, kliknij, kliknij podwójnie, przeciągnij itp.,
  - realizacja rozkazów wieloklawiszowych (kombinowanych) np.  
**Ctrl + Alt + +**      zwiększ rozdzielczość,  
**Ctrl + Alt + -**      zmniejsz rozdzielczość itp.

Praca w trybie tekstowym, preferowana przez wielu doświadczonych informatyków wymaga znajomości rozkazów systemu oraz przynajmniej jednego edytora tekstu, np. vi. Wszystkie rozkazy w linuxie są „wrażliwe” na duże i małe litery (**case sensitive**)

#### 4.3.1. Nazwy urządzeń (device names)

<code>/dev/ fd 0</code>	1 – szy floppy dysk (dyskietka) (to samo co A: w DOS –ie
<code>/dev/ fd 1</code>	2 – gi floppy dysk
<code>/dev/ cdrom</code>	dysk kompaktowy
<code>/dev/ hda 1</code>	1 partycja na 1 – szym dysku twardym podłączonym poprzez kabel IDE (Integrated Device Electronics)
<code>/dev/ hda 6</code>	6 – ta partycja na 1 – szym dysku twardym podłączonym poprzez kabel IDE
<code>/dev/ hdb 1</code>	1 sha partycja na 2 – gim dysku twardym podłączonym poprzez kabel IDE
<code>/dev/ sda 1</code>	1 sha partycja na twardym dysku podłączonym poprzez kabel SCSI (Small Computer System Interface)

#### 4.3.2. Poszukiwanie pliku

Pracując w trybie graficznym można wykorzystywać wyszukiwarę, podobnie jak w systemie Windows i stosować maski (wildkard characters) na oznaczenie części poszukiwanej nazwy (np. gwiazdka \* zastępuje ciąg znaków (ang. string), znak zapytania ? zastępuje pojedynczą literę itd.)

Gdy pracujemy w trybie tekstowym, można wykorzystać polecenia find lub grep opatrzone odpowiednimi parametrami:

`find` punkt startowy -name nazwa\_poszukiwanego\_pliku -print  
na przykład

`find . -name mfp.f -print`      znajdź plik o nazwie mfp.f , rozpoczynając poszukiwanie od bieżącego katalogu

`find / -name h.dat -print`      znajdź plik o nazwie h.dat, rozpoczynając poszukiwanie od katalogu root (/), czyli w praktyce – wszędzie.

`find / -name *readme* -print`   znajdź, ignorując wielkie i małe litery, plik który zawiera ciąg liter (string) `*readme*` rozpoczynając poszukiwanie od katalogu root (/)

jeśli spodziewamy się, że wynikiem poszukiwań będzie długa lista różnych „adresów” poszukiwanego obiektu, celowe jest użycie znaku | less na końcu, który umożliwia stronicowanie tej listy.

`grep ciąg znaków nazwa_zbioru`   znajdź ciąg znaków w zbiorze.

W UNIX – ie panuje zasadniczo „porządek” - wszystkie pliki mają dokładnie określone miejsce w drzewie katalogowym.

Większość katalogów systemu posiada standardowe, stałe nazwy związane z ich zawartością. Poniżej podajemy przykłady tej zawartości

**/** Rdzeń struktury – katalog **root** oznaczony symbolem „/” zawiera plik z nazwami głównych katalogów:

**/bin** zawiera podstawowe programy i polecenia Linuksa, w postaci ładownej (binarnej)

**/boot** zawiera pliki startu

**/dev** zawiera pliki opisu urządzeń komputerowych (fizycznych), rozpoznanych przez LINUX (czyli tzw. sterowniki). Linux traktuje wszystkie dołączone urządzenia po prostu jako pliki.

**/etc** zawiera większość plików konfiguracyjnych, dane chronione oraz niebezpieczne programy usługowe zarezerwowane dla administratora

**/lib** zawiera zbiór bibliotek podprogramów i programów, kompilatorów, w tym również tych niezbędnych do uruchomienia komputera!

**/lost + found** zawiera pliki uszkodzone w razie zawieszenia się komputera

**/misc** pusty, nie wykorzystany katalog

**/mnt** zawiera zamontowane urządzenia:

**/proc** zawiera zbiór informacji o działających aplikacjach i systemie operacyjnym

**/root** katalog domowy administratora systemu. Nie jest to ten sam katalog, co katalog rdzenia struktury **/**. **Należy pamiętać, że / oraz root - to dwa różne katalogi.**

**/sbin** programy binarne do administrowania systemu (dostęp – wyłącznie poprzez rozkaz **su** – (ang. **s**uper **u**ser)

Nikt poza administartorem nie może uruchomić tych programów.

**/spool** – procesy terminowe wywoływane o określonym czasie na przykład z podkatalogów:

- **lpd** – kolejka do drukarki
- **mail** – poczta elektroniczna



`/tmp` miejsce na pliki tymczasowe – każdy może je tam umieścić. Zawartość jest kasowana automatycznie przy starcie komputera.

`/usr` struktura katalogów zawierająca biblioteki, programy, pliki konfiguracyjne dostępne dla wszystkich użytkowników (nie tylko dla **SU** – **Super User**) np.

- `/usr/doc` zawiera dokumentację
- `/usr/games` zawiera gry
- `/usr/man` zawiera manuale
- `/usr/src` zawiera źródła LINUX – a, które może zmieniać użytkownik a także pliki czekające na wydruk
- `/usr/home` – zawiera katalogi i/lub pliki użytkowników, np. `/home/wsosn`

`/var` miejsce przechowywania logów systemowych – plików tekstowych informujących (na bieżąco) o funkcjonowaniu systemu operacyjnego.

Ścieżki (PATH) buduje się tak samo jak w DOS – ie z tą tylko różnicą że:

w systemie DOS elementem rozdzielającym katalogi i podkatalogi jest `\` backslash, zaś w systemie LINUX - `/` slash.

### 4.3.3. Nazwy i rodzaje plików

Przykłady legalnych nazw:

`vi`

`e-konf`

`.bash_history`

`.cshrc`

`.gnome – help – browser`

`gnuplot – 3.7.1 – 12. I 386.rpm` czyli może występować wiele kropek

`pack.tgz`

`mój_zyciorys.txt`

`.bashrc` – plik ukryty linuxa (kropka na początku oznacza plik ukryty)

**Pliki** mogą być:

- wykonywalne (ang. executable): programy, skrypty oraz zbiory danych do programów wykonywalnych
- pliki użytkowników

### 4.3.4. Opcje poleceń

Są oznaczane literami lub ciągiem liter poprzedzonych znakiem minusa (-). Trudno je omawiać ze względu na ogromną liczbę opcji. Zawsze można zaznajomić się z ich bogactwem wypisując na przykład instrukcję: **man nazwa\_polecenia**. Poniżej podajemy jedynie kilka przykładów opcji jednego tylko, wspomnianego już wcześniej polecenia **ls**

**ls -a** wypisuje nazwy wszystkich zbiorów łącznie z nazwami plików ukrytych, rozpoczynających się od ‘ . ‘

**ls -t** sortuje pliki według dat utworzenia

**ls -t -full -time** podaje dokładny czas utworzenia pliku, w pełnym brzmieniu, bez skrótów,

**ls -l** drukuje zawartość katalogu podając kolejno kod dostępu, liczbę powiązań (z innymi elementami twardego dysku), nazwy użytkownika i jego grupy, wielkość zbioru w bitach, czas utworzenia lub ostatniej modyfikacji, nazwa pliku;

Np.: -rw- r-- r-- 1 wsosn zmk 201340 Feb 20 11:20 mfp3d . f

**ls -m** – wypisuje wszystkie nazwy plików w postaci maksymalnie upakowanej, oddzielone przecinkami (tylko nazwy)

**ls -l -n** wypisuje UID (user identity) oraz GID (group identity numbers) (liczby - kody) identyfikacji użytkownika zamiast nazw użytkownika i grupy

**ls -u** sortuje pliki według daty ostatniego dostępu (a nie – utworzenia)

**ls -lr** sortuje pliki w odwrotnej kolejności (czyli np. alfabetycznie od końca)

**ls -lt** sortuje pliki w kolumnie w kolejności czasu ich utworzenia

**ls -ltr** sortuje w kolumnie w kolejności czasu utworzenia „wstecz” (czyli zbiory najstarsze są na początku)

**ls -color** pokazuje pliki w kolorach :

- biblioteki niebieskie
- pliki wykonywalne zielone
- pliki skompresowane, np. \*.rpm czerwone
- linki (połączenia) jasnoniebieskie

### **Zmiana prawa dostępu dla plików i katalogów – polecenie chmod**

Wyróżniamy trzy kategorie użytkowników plików i katalogów: **Właściciel**, **grupa**, **pozostali**.

Prawa dostępu są automatycznie nadawane przez system przy tworzeniu pliku. Prawa te mogą być potem zmieniane, nadawane lub odbierane.

Wyróżniamy zasadniczo trzy prawa: **Czytania**, **pisania**, **wykonywania** . Istnieje jeszcze czwarte **prawo – własności**, o mniejszym znaczeniu, omawiane szczegółowo na przykład w podręcznikach [3, 5, 8].

Każde z wymienionych trzech praw jest definiowane osobno dla każdej z trzech w/w kategorii użytkowników.

Istnieją dwa sposoby oznaczeń praw dostępu: literowe i cyfrowe. Oznaczenia cyfrowe wykorzystujemy w omawianej dalej tzw. bezwzględnej metodzie zmiany prawa dostępu. Oznaczenia literowe są następujące:

r -	prawo odczytu
w -	prawo zapisu
x -	prawo wykonywania
- -	brak prawa

Na przykład na wydruku zawartości katalogu uzyskanym poleceniem `ls -l` można zidentyfikować 10 pierwszych znaków w linijce opisującej plik mfp jako kod dostępu

**-rwx rwx r-x**

```
-rwx rwx r-x 1 wsosn zmk 175863 Now 28 09:25 mfp
```



10 znaków (spacje pomijamy)

plik wykonywalny mfp ma kolor zielony

kod dostępu **-rwx rwx r-x** ma dziesięć znaków

Znak pierwszy:

- (minus) oznacza zwykły plik

d (litera d) oznacza katalog

Pozostałe znaki należy interpretować następująco:

	←	-	←	rwx	←	rwx	←	r-x	←
zwykły	ja	grupa	pozostali						
plik	mam	ma	mogą						
	prawo	te	czytać						
	czytać	same	plik						
	pisać	prawa	i wykonywać						
	wykonywać	co	ale bez						
		ja	prawa zapisu						

**Zmianę prawa dostępu** realizujemy pisząc rozkaz: **chmod** oraz podając (opcjonalnie) kategorię użytkowników oraz argumenty dwu list:

- listy praw dostępu
- listy nazw plików

Listę praw dostępu można zmieniać dwoma sposobami:

- metodą symboliczną
- metodą bezwzględną

Ad a). Metoda symboliczna – zmieniamy prawo dodając plus lub minus przed jego oznaczeniem w polu argumentów polecenia `chmod`, np

```
chmod +x -w nazwa_pliku
```

+ oznacza dodanie prawa (w tym przypadku wykonywanie)

- oznacza zabranie prawa (w tym przypadku pisanie)

Symbolowe prawa można definiować dla poszczególnych kategorii użytkowników podając odpowiedni symbol użytkownika na liście argumentów polecenia `chmod`:

u – dla użytkownika

g – dla grupy

o – dla pozostałych

np.:

`chmod u+rw nazwa_pliku` - oznacza przyznanie prawa czytania i pisania użytkownikowi,

`chmod o+r-wx nazwa_pliku` - oznacza dodanie pozostałym użytkownikom prawa czytania a odebranie prawa pisanie i wykonywania.

Oprócz praw czytania, pisanie i wykonywania można również nadawać prawa własności

Na pozycji prawa wykonywania zamiast `x` pisze się `s`. Symbol ten działa podobnie jak `x`, zmienia tylko nieco prawa do wykorzystania plików (chodzi o prawa własności umożliwiające użytkownikom uruchamianie plików modyfikujących inne pliki właściciela).

Jeśli chcemy, aby po wykonaniu program pozostał w pamięci RAM po zakończeniu jego wykonywania, na pozycji `x` zapisujemy literkę `t` która oznacza tzw. **bit sticky**

Ad. b. Prawa bezwzględne: maski binarne

Metoda bezwzględna jest o tyle wygodna, że zmienia jednocześnie, poprzez podanie jednej 3 – cyfrowej liczby ósemkowej, wszystkie prawa dla wszystkich kategorii użytkowników.

Jest to możliwe ponieważ cyfra ósemkowa odpowiada 3 cyfry binarnym a więc liczba ósemkowa trzycyfrowa może być łatwo przekodowana na  $3 \times 3 = 9$  cyfr binarnych, każda cyfra dla jednego prawa.

Cyfry ósemkowe odpowiadają kolejnym kategoriom użytkowników (właściciel, grupa, pozostali), od lewej do prawej.

Cyfra binarna 1 oznacza prawo przyznane

Cyfra binarna 0 oznacza prawo odebrane

Np. liczba ósemkowa

754

oznacza kod binarny:

1 1 1 1 0 1 1 0 0  
 r w x r - x r - -  
 piszemy więc krótko np. `chmod 754 nazwa pliku`

ósemkowa	binarna
0	000
1	001
2	010
3	011
4	100
5	101
6	110

Cyfry ósemkowe i ich zapis binarny

Np. rozkaz `chmod 700 mój_ważny_plik.for` w zasadzie zabezpiecza go przed dostępem osób niepowołanych - daje mi pełne prawo zapisu, odczytu i wykonywania odbierając jednocześnie to prawo wszystkim pozostałym użytkownikom, zarówno z mojej grupy jak i pozostałym, łącznie z administratorem systemu.

Prawo własności i prawo tzw. bitu sticky nadaje się dodając (z lewej strony) czwartą liczbę ósemkową:

4 (100)      UID    Prawo własności użytkownika  
 2 (010)      GID    Prawo własności grupy  
 1 (001)      bit sticky (pozostawia program w pamięci RAM)

Np. kod prawa dostępu:

4740  $\equiv$  100 111 100 000 oznacza  
 UID rwx r-- ---

czyli nadaje prawo własności użytkownikowi, pozwala użytkownikowi czytać, pisać i wykonywać, grupie czytać a pozostałym odbiera wszystkie prawa dostępu.

#### 4.3.5.Prawa dostępu do katalogów

Prawa dostępu do katalogu definiuje się podobnie jak prawa dostępu do pliku, na przykład podając kod:

`drwxr-xr-x`

pozwalamy wchodzić do katalogu innym użytkownikom, czytać oraz wykonywać pliki tam umieszczone, ale zabraniamy cokolwiek zapisywać w tym katalogu.

Jeśli chcemy, aby takie prawo rzeczywiście zadziało, musimy też udostępnić ścieżkę do katalogu, a więc nadać podobne prawo katalogom nadrzędnym.

### Zmiana właściciela pliku i grupy właścicieli pliku:

Jedynie właściciel pliku może zmieniać prawa dostępu do pliku. Jeśli chcemy nadać innemu użytkownikowi pełną kontrolę nad plikami, możemy przekazać mu prawo własności poleceniem **chown**.

Przykładowo po zastosowaniu tego polecenia do pliku **mfp**:

```
-rwx rwx r-x 1 wsosn zmk 175863 Now 28 09:25 mfp
```

```
chown jurek mfp
```

po komendzie `ls -l` zobaczymy na ekranie

```
-rwx rwx r-x 1 jurek zmk 175863 Now 28 09:25 mfp
```

Podobnie można zmieniać grupę właścicieli plików rozkazem **chgrp**

W praktyce użytkownicy PC-tów występują często w podwójnej roli – jako administratorzy systemu operacyjnego (root) i jako użytkownicy (np. wsosn) i muszą niekiedy formalnie zmieniać prawa własności **własnych** plików (np. root na wsosn lub odwrotnie)

#### 4.3.6. Montowanie urządzeń (dołączanie ich do systemu plików) [3]

W organizacji plików występują tzw. wolumeny wymienne – są to dyski i dyskietki. Pliki mogą być rozmieszczone na różnych urządzeniach pamięciowych, np. na różnych twardej dyskach, CD – ROM – ach, dyskietkach.

Pliki Linuksa umieszczone na poszczególnych urządzeniach (ang. device) są zorganizowane w postaci systemów plików (ułożonych jako drzewo katalogów).

Pozostają oddzielone dopóki nie przyłączy się ich do odpowiedniego "drzewa plików".

Można je przyłączyć czyli zamontować instrukcją `mount`. Urządzenie w Linuksie wygląda więc jako `/dev/ coś`, np. `/dev / cdrom`, `/dev/ floppy` itp.

`/dev/` odpowiada więc literom A:, C:, itp. w systemie operacyjnym DOS. Każde urządzenie jest identyfikowane przez jego sterownik (driver) przechowywany w bibliotece `/dev`.

Pliki może montować tylko użytkownik root (administrator systemu).

Po zamontowaniu dyskietki możemy wykorzystywać rozkazy `mmdir`, `mcopy`, `mdel` i szereg innych, podobnych do odpowiednich poleceń `dir`, `copy`, `del` itd. systemu dos.

#### **mount - urządzenie - miejsce montowania**

np. `mount [ /dev/cdrom ] /mnt/cdrom`

`mount [ /dev/fd0 ] /mnt/floppy`

Część w nawiasach [ ] można pominąć w niektórych instalacjach np. w Red Hat 7.1 i Red Hat 7.2

Jeśli chcemy wymienić dyskietkę w stacji, musimy poprzednią dyskietkę wymontować komendą:

```
umount urządzenie miejsce montowania
```

np. `umount /dev/fd /mnt/floppy` - (czyli montuję konkretną dyskietkę – a nie stację dysków)

lub krócej: `umount /mnt/floppy`

Często te polecenia automatycznie realizuje powłoka graficzna (ang. shell). Gdy nie korzystamy z powłoki, musimy pamiętać o konieczności montażu i demontażu urządzeń.

Demontaż musi być realizowany spoza demontowanego systemu plików. Jeżeli więc jesteśmy zalogowani na CD ROM, bo zrealizowaliśmy rozkazy:

```
mount /dev/mój_katalog mnt/cdrom
cd /mnt/cdrom
```

i prześlemy polecenie:

```
umount /mnt/cdrom
```

to dostaniemy komunikat:

```
umount: /dev/moj_katalog: device is busy
```

sygnalizujący, że nasze polecenie nie zostało wykonane.

Trzeba w tym przypadku przeprowadzić demontaż z katalogu znajdującego się "o szczebel wyżej niż katalog /mnt/, - w tym przypadku jest to katalog root. Dajemy więc polecenie

```
cd .. lub
```

```
cd root
```

a następnie demontujemy płytke CD:

```
umount /mnt/cdrom
```

Dobrze jest przyłączać dyskietki i płytki CD ROM do już zarezerwowanego na te cele katalogu mnt, odpowiednio:

```
/mnt/cdrom      dla dysków kompaktowych CD ROM
/mnt/floppy     dla stacji dyskietek
```

**Skrypty** – są to małe programy budowane dla wygody jako sekwencje kolejno wykonywanych poleceń. Mogą zawierać własne polecenia (definiowane komendą alias). Wygodnie jest przygotowywać je wykorzystując edytor tekstu vi.

**Przykład przygotowania skryptu o nazwie gnutot**, który umożliwi szybką konwersję dużej ilości wykresów sporządzonych wcześniej za pomocą programu gnuplot i zapisanych w plikach o nazwach .gnu na zbiory z rozszerzeniem tex (.tex). Zbiory te mogą być dalej przetwarzane za pomocą profesjonalnego oprogramowania do edycji książek i artykułów TeX lub LaTeX:

1. Wywołujemy edytor vi rozkazem `vi gnutot`

2. Przygotowujemy tekst:

```
gnuplot $1.gnu
mv $1.tex tmp
cat begin tmp end > $1.tex
```

Kolejne linie tego tekstu staną się instrukcjami:

- wywołania programu gnuplot z danymi o nazwie nadanej przez nas o rozszerzeniu gnu
- zmiany nazwy pliku na tymczasową
- połączenia plików begin, pliku o nazwie tymczasowej i pliku end w plik nadający się do przetworzenia programem TeX lub LaTeX, omówionymi na końcu tego rozdziału.

3. Wychodzimy z edytora vi rozkazem `wq` (write and quit)

Otrzymaliśmy plik o nazwie gnutot, który stanie się plikiem wykonywalnym (programem) jeśli nadamy mu prawo wykonywania (+x) rozkazem `chmod`:

```
chmod u +x gnutot
```

W powyższym skrypcie znak dolara \$ zastępuje dowolną nazwę danego wykresu  
Wykresy zrobiliśmy wcześniej programem gnuplot za pomocą rozkazów:

Gnuplot (uruchomienie gnuplot)

```
set term latex
```

```
set output 'nazwa.tex'
```

```
plot 'nazwa' w linesp 1 4
```

'nazwa' oznacza zapisany w kolumnach plik współrzędnych x oraz y wykresu, który chcieliśmy narysować programem gnuplot używając linii typu 1 4, wybranej według opisu tego programu.



Pliki `begin` i `end` występujące w skrypcie `gnutot` zawierają typowe rozkazy otwierające i zamykające edytora TeX lub LaTeX, np. `\begin{document}`, `\end{document}` itp. i mogą być takie same dla wszystkich przetwarzanych wykresów. Plik `tmp` jest plikiem tymczasowym, tworzonym jako kopia tymczasowa kolejnych plików `$1`.

#### 4.3.7. Instalowanie plików `*.rpm`

```
rpm -Ui      xv.rpm          instalacja + uaktualnienie (stara wersja jest usuwana)
rpm -i nazwa.rpm          instalacja pakietu nazwa
rpm -u      nazwa.rpm      usunięcie pakietu nazwa
```

#### Przeniesienie uruchamianego programu „w tło”

`nazwa_polecenia &` → wyświetla nam się nr zadania i procesu oraz pokazuje „znak zachęty” umożliwiający pracę w danym oknie nad innymi zadaniami

#### „Wydobycie” procesu z tła:

```
nazwa_polecenia fg
```

#### Przeniesienie procesu aktywnego „w tło”

```
bg
```

**Archiwizacja danych** (wszystkich danych i podkatalogów zawartych w katalogu o nazwie `nazwa_katalogu`)

```
tar cf nazwa_archiwum.tar nazwa_katalogu
```

Uwaga! archiwizując nie używamy znaku `-` przy opcji `cf` i/lub przy innych opcjach

#### Wydobycie plików z archiwum

```
tar xf nazwa_archiwum.tar
```

#### Kompresja danych i archiwum

```
gzip nazwa
```

```
gzip moje_archiwum.tar
```

#### Dekompresja

```
gunzip nazwa.gz
```

```
gunzip nazwa.tar.gz
```

lub:

```
tar xf nazwa.tgz
```

Również polecenie `compress` pozwala utworzyć skompresowane pliki (z rozszerzeniem `Z`), które można następnie zdekompresować poleceniem `uncompress`.

#### 4.4. Niektóre kłopoty z Linuxem i/lub Unixem

Przykłady:

INVALID USERNAME OR PASSWORD - trzeba określić hasło i nazwę użytkownika (ang. username) – nadać je przy instalacji lub uzyskać od administratora.

START X nie uruchamia GUI (Graphic User Interface) GUI jest opcjonalne w UNIXie – niekoniecznie musi być zainstalowany.

GNOME . lub inny GUI daje „wycinkowy” obraz ekranu. Zmienić rozdzielczość na przykład z 800 x 600 lub 640 x 480 na 1024 x 768 lub inną naciskając CTRL + ALT i jednocześnie + i/lub - (\* w Windows ten sam efekt uzyskiwaliśmy poprzez Panel Sterowania → Ekran → Ustawienia).

Sporządzenie dysku startowego (Ang. *Emergency* Boot Disc – Dysk „awaryjnego” startu)

- A. montujemy czystą dyskietkę
- B. uname – r                   określamy wersję Linuxa
- C. mkbootdisk -device /dev/fd0 nr\_wersji                   podajemy nr\_wersji Linuksa  
uzyskany jako odpowiedź w punkcie B.

##### 4.4.1. Podłączenie do internetu

- A. Do sieci bezpośrednio – poprzez ETHERNET
- B. Przyłączamy się jako “klient” poprzez modem korzystając z połączenia telefonicznego. Musimy utworzyć, podobnie jak w Windows, tzw. dial-up account.  
Potrzebny jest modem obsługujący linuksa i Dostawca Usług Internetowych (Internet Service Provider (ISP)) z serwerem. W przypadku wprowadzanych ostatnio usług internetowych typu Neostrada trzeba wykonać szereg instrukcji opisanych w manualu modemu sprzedawanego w ramach tej usługi.

Przykład niektórych danych potrzebnych do skonfigurowania połączenia internetowego:

- |  |  |
|--|--|
| 1. Nr telefonu ISP (serwera)                                     | 8264396  |
| 2. Username  | wsosn  |
| 3. Hasło (Login password)  | *****  |
| 4. DNS (ang. Domain Name Serwer)                                 | możemy uzyskać ten numer pisząc np.<br>host brama.ippt.gov.pl → uzyskamy informację, np: |
|  | 148. 81. 52. 11  |
| 5. Name Server DNS 1   | 148. 81. 52. 11  |
| 6. Name Server DNS 2   | 148. 81. 55. 199   |
| 8. IP address (jeśli nie dynamiczny)                             | IP = Internet protocol 148 – 81 – 53 -<br>206  |
| 9.   | 255 – 255 – 252 – 0  |
| 10. E – mail adres   | <a href="mailto:wsosn@ippt.gov.pl">wsosn@ippt.gov.pl</a>                                 |
| 11. Incoming (POP 3) mail server<br>(server poczty przychodzącej | brama.ippt.gov.pl  |
| 12. Outgoing (SMTP) mail serwer                                  | brama.ippt.gov.pl  |
| 13. Email user name  | wsosn.ippt.gov.pl  |
| 14. Email Password   | ****   |

Dostawcy usług internetowych podają zwykle wszystkie szczegóły wymagane do skonfigurowania połączenia poprzez konkretny modem zalecany przez danego dostawcę.

#### 4.4.2.Dobór modemu

Większość zewnętrznych modemów może być wykorzystana do ustawienia połączenia.

Większość modemów wewnętrznych **nie obsługuje linuxa**.

Rozkaz identyfikujący port przyłączenia

startserial /dev/cua0      lub:    start serial /dev/cua1    itd.

Trzeba znaleźć taki port (najlepiej pod WINDOWS) , że w/w rozkazie w liniach zobaczymy informację:

RTS	1	Request to Send
CTS	1	Clear to Send
DSR	1	Data Set Ready

Jeśli mamy takie połączenie, to musimy skonfigurować modem rozkazem `modemtool` uaktywniamy katalog `/dev/modem/` (podobnie jak to było w omawianym wcześniej przypadku katalogów `/dev/cdrom/` lub `/dev/fd0/`)

### **Pliki zawierające ustawienia systemowe**

Mogą być wprowadzone modyfikowane tylko przez osoby uprawnione, ale dodatkowo można je ukryć przed okiem innych użytkowników.

Ich nazwa bywa poprzedzona kropką, np:

<code>.bash_history</code>	<code>.gimp</code>
<code>.cshrc</code>	<code>.gnome</code>

`.bashrc` - jest to plik, w którym można zdefiniować własne polecenia komendą `alias`.

Tak naprawdę pliki te nie są zbyt mocno schowane, można je łatwo wykryć np. komendą `ls -a`.

### **Zbiory ukryte pod interfejsem graficznym (powłoką)**

Zbiory ukryte pod powłoką np. XWindows lub Gnome, można wykryć wybierając z Menu odpowiednie polecenia myszą.

Kolejno “klikamy”: File Manager > Edit → Preferences → Show Hidden Files → OK. mouse button.

Zbiory ukryte pokazują się jako pierwsze za bibliotekami.

### **Uruchamianie programów w trybie tekstowym,**

- po prostu piszemy nazwę programu, poprzedzoną niekiedy ścieżką, np.:

`./nazwa_programu_wykonywalnego < dane > wyniki`

`bc` (uruchamianie kalkulatora).

`f77 -w -o nazwa_programu_wykonywalnego nazwa.f` - kompilacja programu `nazwa.f` napisanego w fortranie. W niektórych dystrybucjach linuxa można używać do tego celu kompilatora `g77` umożliwiającego również kompilację programów przygotowanych w języku C. W Aurox – e kompilator fortran obsługuje programy napisane w Fortranie 90.

### **4.4.3.Pomoc w LINUXIE**

- A. strony man (od ang. manual czyli poradnik, instrukcja użytkownika)  
pod GUI “klikamy” Gnome panel → Integrated Help System
  - B. strony x man (alternatywne informacje, nieco inaczej o tym samym)
  - C. whatis polecenie np. whatis Ls
  - D. info polecenie np. info su (super user)
  - D. howto - trzeba szukać na dysku instalacyjnym, normalnie nie instaluje się tych dokumentów. Jest ich bardzo dużo.
- Aby dostać się do tych informacji, trzeba zamontować dysk instalacyjny i czytać z /mnt/cdrom /doc/HOWTO directory Przykładowo informacje o myszy trzyprzyciskowej znajdujemy:

TEMAT	<u>GDZIE ZNALEŹĆ INFORMACJE</u>
3 Button Mouse	doc   HOWTO   mini
podobnie:	
Dos – to – Linux – HOWTO	doc   HOWTO
IP – Alias	doc   HOWTO   mini
LILO	doc   HOWTO   mini
Partition	doc   HOWTO   mini
Polisch HOWTO	doc   HOWTO
TeX	doc   HOWTO

Aby znaleźć program, możemy użyć omówionej wcześniej instrukcji find lub locate np.  
locate nazwa | less

Aby zainstalować konkretny program, trzeba wejść do pakietu i zajrzeć do zbioru | usr | doc

- E. w sieci WWW – np na stronie  
[www.redhat.com/mirrors/laptop/welcome.html](http://www.redhat.com/mirrors/laptop/welcome.html)
- F. News groups and Mailing list. Korzystanie z tej formy zdobywania informacji wymaga przestrzegania pewnych zasad - wypada, aby odpowiedzieć na czyjeś pytanie jeśli oczywiście sami znamy jakąś rozsądną odpowiedź. W zasadzie nie zdarza się, aby nasze pytanie pozostało bez odpowiedzi.

Określenie rzeczywistych rozmiarów katalogów - pod GUI (standardowo pokazywany rozmiar – np. 4096, nie odpowiada rozmiarowi rzeczywistemu)  
→ zaznaczamy plik – nazwę katalogu  
→ wybieramy opcję → plik → wyświetlaj rozmiary katalogów

## Niektóre funkcje powłoki

Powłoka wspomniana już w podpunktach 4.2 i 4.3 jest interpreterem poleceń udostępniającym zorientowany wierszowo, interaktywny interfejs pomiędzy użytkownikiem i systemem operacyjnym (SO). Dzięki takim interpretacyjnym zdolnościom powłoka może bardzo ułatwić przekazywanie poleceń do SO, ukierunkowywać strumień wyników, generować nazwy plików, wykonywać operacje w tle itd.

Tworzenie skrótów do programów, które często uruchamiamy, np. User Mount Tool:  
Gnome → Programy → System → Disk Management → : Prawy przycisk myszy

Ukazuje się ikona - prawa stopa ← Dodaj ikonę uruchomieniową do panelu

### Ukierunkowanie

Np. chcemy wynik komendy ls skierować nie na ekran a do pliku. Piszemy  
\$ cat ls > zawartość katalogu

Dołączanie pliku do zbioru plików

\$ cat mój\_skrypt wszystkie skrypty

Ukierunkowanie standardowego wejścia: tworzymy plik o nazwie moje\_dane

\$ cat > moje\_dane

dane1

dane2

dane3

itd...

zakończenie ^D

^D

(^D kończy wpisywanie)

rozkaz cat moje\_dane pozwala obejrzeć wynik naszej pracy

### Zmienne powłoki : =, \$

Sposób ich wykorzystania ilustrują sekwencje rozkazów w poniższych ramkach

```
$ student =Kowalski
```

```
$ echo $ student
```

```
Kowalski
```

```
$ echo student
student
```

Czyli znakiem = przypisujemy wartość Kowalski do zmiennej student  
\$ jest operatorem odnoszącym nazwę zmiennej do jej wartości:

### Albo

```
$ wykonanie = /home / wsosn / dim 3 / . / mfp
$ wykonanie < dane > wyniki
pwd
/home / wsosn
```

Można umieszczać polecenia powłoki wewnątrz pliku a następnie spowodować, aby powłoka odczytała i wykonała polecenia z pliku. Plik staje się więc programem. Taki plik nazywa się **skryptem powłoki**.

Sekwencję poleceń w skrypcie można przygotować korzystając ze standardowego edytora tekstu, np. Vi, lub wykorzystując omówione wyżej polecenie cat, na przykład:

```
cat > mój_skrypt
date
ls
^D
```

Tak przygotowany skrypt mogę uruchomić rozkazem sh lub . (czyli kropką!)

```
sh mój_skrypt
lub
```

```
. mój_skrypt
```

spacja

Mogę też nadać temu plikowi prawo wykonywania rozkazem chmod u + x mój\_skrypt\_i wtedy mój\_skrypt stanie się wykonywalnym programem uruchamianym rozkazem ./mój\_skrypt

Wszystkie polecenia LINUXa mogą przyjmować argumenty. Dotyczy to również skryptów.

Do argumentów polecenia odwołujemy się kolejno cyframi 1,2,3 itd. Odwołanie następuje przy użyciu operatora \$.

Np.

```
Cat >testdolar
```

```
ls /home/wsosn/*. $1
```

```
^D
```

Tak utworzony skrypt testdolar pozwoli na przykład wyświetlać wszystkie pliki o rozszerzeniu rpm z katalogu /home/wsosn/ (choć jesteśmy w innym katalogu), rozkazem `./testdolar rpm`

### **Anulowanie zadań**

`$ jobs` wyświetla aktualnie wykonywane zadania, np.

```
[ 1 ] +   Running   mfp
[ 2 ] -   Running   Lpr *.f
```

`$ kill % 2` anuluje zadanie 2

(w w/w zadaniach `+` oznacza że zadanie jest właśnie wykonywane, zaś `-` oznacza że zadanie czeka w kolejce do wykonania).

### **Przerywanie zadań ^Z**

Np.

```
$ ./ moje zadanie
```

```
^ Z
```

```
$ bg
```

```
$ jobs
```

```
[ 1 ] + stopped moje zdanie
```

Wznawianie zadań fg, np. dla w/w. zadania wznowienie następuje po komendzie

```
$ fg % 1
```

### **Ustalenie daty i czasu**

```
$ date
```

### **Komunikacja - polecenia talk i mail**



talk ulrika – jeżeli ulrika jest zalogowane na wspólnym serwerze możecie pisać do siebie jednocześnie!

mail – uruchomienie standardowego programu pocztowego zorientowanego wierszowo

```
$ mail Cristina
```

```
Subject: obiad
```

```
Umówmy się na 14:00
```

```
OK.?
```

```
^D
```

```
EOT
```

```
$
```

```
$ set record = „wychodzące”
```

Czytanie poczty: mail nr z listy listów

p \$ - wyświetlanie ostatniego listu

p ^ -3 – wyświetlanie listów od 1 do 3 itd.

R – respond to

q – wyjście z zachowaniem listu nie przeczytanego

x – wyjście z zachowaniem wszystkich listów łącznie z usuniętymi  
itd.

s -save – zapisanie listu jako pliku do jakiegoś katalogu

```
s 2 moje listy
```

itd.

W wielu dystrybucjach linuxa występuje zorientowany ekranowo program pocztowy Elm

#### 4.5. Wybrane polecenia administratora systemu

Jak już pisaliśmy katalog `/sbin` zawiera programy binarne do administrowania systemu. Dostęp do tego katalogu jest możliwy po zalogowaniu się do systemu jako administrator (root) lub z konta dowolnego użytkownika - wyłącznie poprzez rozkaz **su** – (ang. **s**uper **u**ser) Nikt poza administratorem nie może uruchomić tych programów. Powyższe ograniczenie dotyczy wielu innych katalogów (np. katalogu `/etc` zawierającego pliki konfiguracyjne)

Uruchamiając przykładowo program

```
/etc/lilo.conf
```

można ustalić domyślnie (ang. by default), jak będzie ładował się linux lub dos, w jakiej kolejności itd. Edycja pliku lilo.conf umożliwia wprowadzenie zmian.

**Niektóre stany systemu** (ustawiane poleceniami z katalogu `u/sbin/init`)

```
init 0 - zamyka system
```

```
init 1 - daje dostęp do systemu jedynie dla superusera (root-a)
```

```
init 3 - umożliwia dostęp wielu użytkowników
```

init 6 - zatrzymuje system i uruchamia go od nowa

### Zwykle większe systemy zamykane są poleceniem

```
$ shutdown -r now natychmiastowe zamykanie systemu
$ shutdown -h 10 zamykanie systemu po 10 minutach
$ shutdown -h 5 „system wymaga restartu”
```

Rozkazom towarzyszy informacja dla użytkowników.

### Dodawanie użytkowników

w bibliotece /usr / bin wykonujemy polecenie

```
$ adduser iza
```

a następnie wprowadzamy odpowiednie dane w trybie konwersacji z komputerem

```
Loking      for first available  VID .....505
Loking      for first available  GID .....505
Adding      login: iza ... done
Creating    home directory:  /home/ iza ...done
Creating    mailbox:        /var/spod/mail/iza ...done
Do not forget to set the password.
$ Passwd iza .....
```

Podobnie dodajemy grupy użytkowników. W nowszych dystrybucjach linuxa można znacznie łatwiej dodawać użytkowników korzystając z systemu menu w trybie graficznym.

## 4.6. Edytor vi

Szczegółowy opis tego edytora, standardowo dołączanego do każdej dystrybucji linuxa, można znaleźć na omawianych wyżej stronach man, pisząc po prostu **man vi**

Istnieją dwa tryby pracy z tym bardzo wygodnym i „szybkim” edytorem:

A: tryb poruszania się po ekranie i operacji „ekranowych” oraz

B: tryb pracy z tekstem („pisanie”)

Tryby pracy zmieniamy następująco:

Po wyłączeniu edytora `vi` nazwa mamy tryb ekranowy, A:

Aby zacząć pisać albo dopisywać, piszemy literę `i` (od ang insert) lub `a` ( od angielskiego add). Na dole ekranu ukazuje się napis INSERT. Powrót do trybu ekranowego następuje poprzez naciśnięcie klawisza ESC.

Wyjście z edytora:

Musimy przejść do trybu ekranowego (nie możemy wyjść z edytora w trybie tekstowym)!

Piszemy znak dwukropka:

(na dole pojawia się znak: )

oraz `q` - wyjście bez zapisywania (`q` - wyrazem wyjścia bez zapisu)

`:q` nie pozwoli nam wyjść łatwo z tekstu, w którym dokonaliśmy już jakichś zmian - w tym przypadku możemy napisać `wq` ( ang write and quit). Jeśli chcemy zrezygnować z wprowadzonych zmian, musimy użyć `q!` aby wyjść z edytora.

`qw` - wyjście z zapisaniem

`w` - zapis bez wyjścia

[ Dwa wykrzykniki przy pracy w trybie tekstowym oznaczają ponowne wejście do tekstu (lub klawisz "do góry")]

`yy` - zapamiętanie wiersza w buforze

`p` - wczytanie w/w wiersza czyli wprowadzenie go z bufora do tekstu

`x yy` - zapamiętanie x linii w buforze

`p` - jw. – wczytanie owych x linii

`dd` - wykasowanie linii

`x dd` - wykasowanie x linii

UWAGA: x w powyższych przykładach oznacza dowolną liczbę.

`^G` Określenie numeru porządkowego linii

`^G` używamy też aby wyciąć dłuższy fragment programu (zamiast `x dd` wymagające określenia x)

ustawiamy kursor na początku wycinanego fragmentu, określamy nr linii początkowej; `xp` polem na końcu, określamy nr linii końcowej; `xk` a następnie, znając `xp` i `xk`, wypisujemy rozkaz:

`xp, xk` w nazwa fragmentu

zapisujemy ten fragment w katalogu w którym aktualnie pracujemy.

Inne użyteczne rozkazy:

nr linii `shift G` idziemy do linii nr linii

`:r nazwa fragmentu` – wczytanie do tekstu zbioru `nazwa fragmentu`.

(string) / ciąg znaków - znalezienie ciągu znaków (trzeba pamiętać że w UNIXie, LINUXie małe i duże litery oznaczają co innego!

`x yL` zapamiętanie w buforze x znaków

`p` wyjęcie z bufora wyżej wymienionych x znaków

/ ciąg znaków powoduje szukanie „w dół” tekstu ciągu znaków

n powoduje następne szukanie tego ciągu znaków  
 Shift n - szukanie „w górę” tekstu  
 r (od ang. replace) (nowy znak) zmiana znaku na inny, nowy znak  
 x – kasowanie znaku  
 {liczba znaków n}x - kasowanie n znaków  
 u – odwrócenie kasacji (powrót do starego tekstu)

#### 4.7. Programy TEX i LATEX

Są to w pełni profesjonalne programy doskonale nadające się do pisania książek, artykułów i innych prac zawierających liczne wzory, rysunki, bogatą bibliografię itp. Umożliwiają bardzo wygodną współpracę z bazami danych zawierającymi na przykład literaturę przedmiotu, zbiory danych do obliczeń, zbiory rysunków, zdjęć. Pozwalają na łatwą modyfikację przygotowywanego tekstu zmieniając automatycznie, w przypadku poprawek, numerację rozdziałów i podrozdziałów, wzorów, rysunków czy tabel. Użytkownik tych kodów może mówić o programowaniu tekstu – którego „obróbka” kończy się kompilacją prowadzącą do wygenerowania pliku nadającego się do wydruku. Omówienie programów przekracza zakres niniejszego opracowania. Na rynku jest wiele książek opisujących TeX i LaTeX, np. [10].

Poniżej podajemy przykład skryptu (czyli programu sterującego) napisanego przy pomocy edytora **vi** i służącego do kompilacji tekstu w TeX-u lub LaTeX – u. Pliki **Wstęp**, **mono1**, **mono2**, **zal** itd. zawierają tekst książki, którą przygotowaliśmy dla Wydawnictwa. Baza danych o nazwie **biblio.bib** zawierają spis literatury. Rozszerzenie nazwy **biblio.bib** jest standardowe i dlatego nie występuje w instrukcji. Znak **%** w jednej z instrukcji oznacza jej wyłączenie. Instrukcja ta jest więc w tym przypadku traktowana przy kompilacji tak jak wiersz komentarza.

```
\begin {dokument}
\ chapter{Wstęp}
\cite {Czochralski}
\input{mono1}
\input{mono2}
\input{mono3}
%\input{zal}
\bibliography{biblio}
\bibliographystyle{plain}
\tableofcontents

\addtocontents{toc}{\protect\vspace{2ex}}
\addtocontents{toc}{{\bf Spis tre\'sci \hspace{11cm} 150 - 152}}

\end{document}
```

#### 4.8. Literatura

- [1] Encyklopedia multimedialna 2002 (na płycie CDROM)
- [2] **Płoski Z.:** Informatyka, komputer i Internet, Słownik encyklopedyczny, Wyd. Europa, Wrocław 2001
- [3] **Petersen R.:** Arkana: Linux, Wydawnictwo RM, 1998
- [4] Instalacja systemu Aurox Linux 9.4, Linux +Extra, AUROX, czerwiec 2004
- [5] **Simpson A., Ray J.:** Using Redhat Linux, special edition, QUE, Indianapolis, USA, 2000
- [6] Czasopismo Linux Plus, strony www, np. [www.aurox.pl](http://www.aurox.pl) , [www.redhat.com](http://www.redhat.com) , [www.linux.org.pl](http://www.linux.org.pl)
- [7] **Kleger – Radomin R.:** Dyplom spod Latexa, Linux +, nr 5/2001 (49) MAJ
- [8] **Podstawczyński A.:** Linux, praktyczne rozwiązania, Helion 2000
- [9] Edytor VI, strony man systemu Linux
- [10] Lesli Lamport, LaTeX - A Document Preparation System, Addison-Wesley Publishing Company, 1986