

Temat zajęć: Wprowadzenie oraz obsługa systemu plików.

<i>Czas realizacji zajęć:</i>	135 min.
<i>Zakres materiału, jaki zostanie zrealizowany podczas zajęć:</i>	Logowanie do systemu, obsługa pomocy systemowej, obsługa systemu plików, tj. tworzenie, usuwanie, zmiana nazwy i przenoszenie plików oraz katalogów, struktura katalogów, wyszukiwanie plików i katalogów, prawa dostępu, dowiązania.

I. Wstęp oraz logowanie do systemu.

Systemy operacyjne Unix są systemami *wielodostępnymi* i *wielozadaniowymi* – wielodostępność pozwala na jednoczesną pracę wielu użytkowników, a wielozadaniowość umożliwia uruchamianie w systemie wielu zadań jednocześnie. Wielodostępność wymaga stosowania mechanizmów, które pozwalają na bezpieczną i jednoczesną pracę różnym użytkownikom. Podstawowym rozwiązaniem w tym zakresie jest *autoryzacja*, która jest realizowana w oparciu o nazwy użytkowników i ich hasła. Rozpoczęcie pracy z systemem operacyjnym wymaga zatem podania tych dwóch informacji (Rysunek 1).

```

Welcome to Unix (i386) - Kernel 2.6.16-1.2122 (tty1)

login: adam
passwd:

Last login: Fri Jun  9 12:56 - 13:10  (00:13) tty1
Welcome to unix.test.net

[adam@unix ~]$

```

Rysunek 1: Logowanie do systemu Unix

Po zalogowaniu do systemu użytkownik ma możliwość zmiany własnego hasła – służy do tego polecenie:

```
passwd
```

Po jego wydaniu należy wprowadzić dotychczasowe hasło, a następnie nowe hasło, które trzeba także powtórzyć, aby system mógł zweryfikować czy nie popełniono błędu przy jego wprowadzaniu (np. “literówki”). W trakcie wpisywania hasła na monitorze nie są wyświetlane żadne znaki. System identyfikuje użytkowników na podstawie specjalnych identyfikatorów numerycznych (takie numery są łatwiejsze do przetwarzania) – nazywanych *id*. Każdy użytkownik systemu operacyjnego może także sprawdzić jaki identyfikator numeryczny *id* został mu przydzielony – należy wówczas zastosować polecenie:

```
id
```

którego wynik może być następujący:

```
uid=1500(adam) gid=100(students) groups=100(students)
```

gdzie **uid** (ang. *user identifier*) oznacza identyfikator użytkownika, w nawiasie znajduje się jego nazwa, **gid** (ang. *group identifier*) oznacza identyfikator grupy – każdy użytkownik należy do pewnej grupy, tzw. *grupy podstawowej* (dla powyższego przykładu jest to grupa o identyfikatorze 100 i nazwie *students*), dodatkowo każdy użytkownik może należeć do innych grup, są one wymienione po słowie **groups**.

Wielodostępność oznacza także, że każdy użytkownik może zalogować się do systemu wielokrotnie, korzystając np. z *terminali wirtualnych* dostępnych za pomocą klawiszy **Alt-F1**, **Alt-F2** itd. lub **Ctrl-Alt-F1**, **Ctrl-Alt-F2** itd. Możliwe jest także pozyskanie informacji o wszystkich zalogowanych aktualnie w systemie użytkownikach – aby tego dokonać należy wydać

jedno z poleceń:

who
finger

Wylogowanie się z systemu jest możliwe po wydaniu polecenia:

exit

Warto także zaznaczyć, że systemach UNIX rozróżniane są wielkie i małe litery: dotyczy to zarówno logowania do systemu jaki i wydawania poleceń w systemie oraz nazw plików i katalogów.

II. Pomoc systemowa.

W systemach Unix dostępna jest pomoc systemowa w postaci dokumentów tekstowych opisujących różne aspekty systemu i jego narzędzia. Dostęp do dokumentacji możliwy jest dzięki przeglądarce interaktywnej, którą uruchamia się poleceniem:

man nazwa_polecenia

gdzie *nazwa_polecenia* to nazwa programu lub usługi, dla której chce się uzyskać pomoc, np.:

man passwd

pozwała na uzyskanie pomocy dla polecenia **passwd**. Istnieją także strony pomocy dla samej pomocy systemowej – można uzyskać do nich dostęp poleceniem:

man man

Strony dokumentacji podzielone są na sekcje, które separują informacje zawarte w pomocy – oto lista najpopularniejszych sekcji:

- NAME – nazwa oraz krótki komentarz lub wyjaśnienie;
- SYNOPSIS – sposoby uruchamiania programu lub polecenia wraz z listą możliwych przełączników;
- DESCRIPTION – pełen opis programu oraz szczegółowy opis możliwych do zastosowania przełączników;
- CONFIGURATION – opis konfiguracji usługi lub programu;
- FILES – opis plików konfiguracyjnych;
- SEE ALSO – wskazówki dotyczące podobnych lub powiązanych poleceń.

Całość dokumentacji podzielona jest na rozłączne rozdziały, które zawierają opisy poleceń i programów określonego typu. Podział na rozdziały może się różnić w zależności od implementacji systemu, jednak najczęściej podział wygląda następująco:

- rozdział nr 1 – polecenia użytkownika;
- rozdział nr 2 – wywołania systemowe;
- rozdział nr 3 – funkcje biblioteczne;
- rozdział nr 4 – pliki specjalne;
- rozdział nr 5 – formaty plików;
- rozdział nr 6 – gry;
- rozdział nr 7 – konwersje i różnice;
- rozdział nr 8 – administracja i polecenia administratora;
- rozdział L – funkcje bibliotek matematycznych;
- rozdział N – funkcje tel.

Strony pomocy są oznaczane za pomocą hasła i numeru rozdziału, np.:

passwd(1)

oznacza, że dla polecenia użytkownika **passwd** pomoc systemowa znajduje się w rozdziale nr 1. Przywołanie pomocy ze wskazaniem rozdziału wygląda następująco:

man 1 passwd

Takie polecenie wyświetli pomoc systemową dla polecenia **passwd** z rozdziału nr 1.

Pomoc systemowa wyświetlana jest za pomocą przeglądarki **more**, którą obsługuje się za pomocą następujących poleceń klawiszowych:

- **spacja** – przejście do następnej strony;

- **Ctrl+B** – przejście do poprzedniej strony;
 - **q** – zamknięcie i opuszczenie przeglądarki;
 - **/** – wyszukiwanie tekstu w przód, po znaku **/** należy wpisać tekst do wyszukania;
 - **?** – wyszukiwanie w tył;
 - **n, N** – przejście do następnego (**n**)/poprzedniego (**N**) wystąpienia poszukiwanego wyrażenia.
- Wyszukiwanie stron pomocy systemowej jest możliwe dzięki programom **apropos** oraz **whatis**, które wyszukują podanych słów w pomocy systemowej, np.:

```
apropos passwd
whatis passwd
```

III. Wydawanie poleceń.

Polecenia w systemach Unix wydaje się wpisując nazwę polecenia oraz wybierając klawisz **Enter**. Polecenia mogą przyjmować *argumenty* oraz *przełączniki*, zgodnie ze wzorcem:

```
polecenie [przełączniki] [argumenty]
```

Oto przykład:

```
ls -al /etc
```

W tym przypadku polecenie **ls** zostało uruchomione z argumentem **/etc** oraz z przełącznikami **al**. Wydanie polecenia **ls** bez argumentu spowoduje wyświetlenie zawartości aktualnego katalogu; podanie natomiast jako argumentu nazwy katalogu, spowoduje wyświetlenie zawartości tego, wskazanego katalogu – w przykładzie jest to katalog **/etc**. Dalej, domyślne działanie polecenia można modyfikować stosując przełączniki, które są przeważnie jednoznakowymi skrótami, zawsze występującymi po znaku “-”. W przykładzie zastosowano dwa przełączniki **a** i **l**. Opis możliwych do zastosowania argumentów i przełączników można odnaleźć w pomocy systemowej dla właściwego polecenia. Przełączniki, jak widać, można łączyć – kolejny przykład przedstawia jak można wydawać polecenie z przełącznikami:

```
ls -a -l /etc
ls -l -a /etc
ls -al /etc
ls -la /etc
```

IV. Struktura katalogów.

Katalog jest strukturą, która umożliwia porządkowanie i grupowanie danych, jakie są przechowywane na dysku komputera. Katalogi mają strukturę hierarchiczną – w każdym katalogu mogą być przechowywane zarówno pliki, jak i katalogi niższych poziomów. W systemach UNIX wszystkie dostępne systemy plików postrzegane są przez użytkownika jako jedna struktura katalogów, której wierzchołkiem jest tzw. *katalog główny*, oznaczany symbolem **/**. Jeśli w systemie dostępnych jest kilka systemów plików to jeden z nich jest systemem głównym, a pozostałe są do niego *montowane*. Regułą jest także, że każdy użytkownik posiada tzw. *katalog domowy*, zwykle jest to podkatalog katalogu **/home**, w którym użytkownik może przechowywać własne pliki.

```

/
├─ bin
├─ boot
├─ dev
├─ etc
├─ home
├─ lib
├─ lost+found
├─ media
├─ mnt
├─ opt
├─ proc
├─ root
├─ sbin
├─ srv
├─ sys
├─ tmp
├─ usr
└─ var

```

Rysunek 2: Przykładowa struktura katalogów systemu Unix

Postać systemu plików systemów UNIX (Rysunek 2) jest w dużej części ustandaryzowana – oto najistotniejsze podkatalogi katalogu głównego:

- `/bin` – katalog zawierający programy niezbędne do uruchomienia systemu;
- `/dev` – katalog zawierający pliki specjalne, które reprezentują dostępne urządzenia;
- `/etc` – katalog z lokalnymi plikami konfiguracyjnymi systemu;
- `/home` – w tym katalog znajdują się podkatalogi domowe użytkowników systemu;
- `/proc` – wirtualny system plików, który dostarcza informacji o bieżących procesach w systemie i jego jądrze;
- `/root` – zwyczajowo katalog domowy użytkownika `root`, czyli administratora systemu;
- `/usr` – katalog zawierający zestaw oprogramowania użytkowego dostępnego dla użytkowników;
- `/var` – katalog ten zawiera pliki, które często zmieniają swoją zawartość i/lub rozmiar.

Dokładniejszy opis struktury katalogów systemu dostępny jest w pomocy systemowej `hier(7)` – aby uzyskać do niego dostęp należy wydać polecenie:

```
man 7 hier
```

Informacja o katalogu bieżącym jest wyświetlana po wydaniu polecenia:

```
pwd
```

Jak wspomniano symbol `/` oznacza katalog główny, używa się także innych symboli dla określenia wybranych katalogów:

- `.` – oznacza katalog bieżący,
- `..` – oznacza katalog bezpośrednio nadrzędny;
- `~` – oznacza katalog domowy użytkownika.

V. Obsługa katalogów.

Podstawowe operacje obsługi katalogów można realizować z wykorzystaniem następujących poleceń:

- `cd [przełączniki] nazwa_katalogu` – zmiana katalogu, np.:
`cd /etc` – zmienia katalog bieżący na katalog `/etc`;
`cd ~` – zmienia katalog bieżący na katalog domowy użytkownika;

- `cd ..` – zmienia katalog bieżący na katalog bezpośrednio nadrzędny.
- `ls [przełączniki] [nazwa_katalogu]` – wyświetlenie zawartości katalogu, np.:
 - `ls` – wyświetla zawartość katalogu bieżącego;
 - `ls -a` – wyświetla zawartość katalogu bieżącego uwzględniając wszystkie pliki – tzn. także te, których nazwa zaczyna się od znaku “.” (umownie są to pliki ukryte);
 - `ls -al` – wyświetla wszystkie pliki z katalogu bieżącego z uwzględnieniem tzw. “długiego formatu”, czyli podając typ każdego obiektu w katalogu (pierwszy znak linii: d – katalog, znak “-” – plik zwykły, l – dowiązanie), prawa dostępu, liczbę dowiązań, właściciela, nazwę grupy, rozmiar (w bajtach), data ostatniej modyfikacji oraz nazwę (Rysunek 3);
 - `ls -al ~` – jak wyżej, przy czym wyświetlana jest zawartość katalogu domowego;
 - `ls -al /etc` – jak wyżej, ale wyświetlana jest zawartość katalogu `/etc`.

```
[adam@unix temp]$ ls -al
drwxrwxr-x  3 adam students 4096 lip  2 23:55 .
drwxr-xr-x 51 adam students 4096 lip  2 22:12 ..
drwxr-xr-x  9 adam students 4096 cze 30 10:30 abc
-rwxr-xr-x  1 adam students 1751 cze 30 11:32 plik.txt
```

Rysunek 3: Przykładowy wynik działania polecenia `ls -l`

- `mkdir [przełączniki] nazwa_katalogu` – tworzenie katalogów, np.:
 - `mkdir ./xyz` – utworzenie katalogu `xyz` w katalogu bieżącym;
 - `mkdir ../xyz` – utworzenie katalogu `xyz` w katalogu bezpośrednio nadrzędnym.
 - `rmdir [przełączniki] nazwa_katalogu` – usuwanie katalogów, np.:
 - `rmdir ~/xyz` – usunięcie katalogu `xyz` z katalogu domowego;
 - `rmdir ./xyz` – usunięcie katalogu `xyz` z katalogu bieżącego.

Dla poleceń `rmdir` i `mkdir` dostępny jest m.in. przełącznik `-p`, który pozwala odpowiednio, usuwać i tworzyć struktury katalogów, np.:

 - `rmdir -p abc/def/ghi` – usunie katalogi `ghi`, `def` oraz `abc`, które tworzyły hierarchię.

VI. Obsługa plików.

Plik to zdefiniowana (przeważnie przez użytkownika) porcja danych, która jest przechowywana w systemie w pamięci masowej. W systemach UNIX niemal wszystko jest plikiem, także urządzenia są reprezentowane przez *specjalne pliki*. Pozwala to, na zachowanie spójnego sposobu dostępu i obsługi do wielu heterogenicznych zasobów, w jeden transparentny sposób. Nazwy plików nie mają podziału na nazwę i rozszerzenie, jednakże można takie podejście stosować; możliwe jest stosowanie w nazwach plików znaków specjalnych (np.: \$, % lub #), ale nie jest to zalecane.

Podstawowe operacje obsługi plików można realizować z wykorzystaniem następujących poleceń:

- `cp [przełączniki] nazwa_pliku nowa_nazwa_lub_katalog` – kopiowanie pliku określonego przez pierwszy argument pod nazwą lub do katalogu określonego drugim argumentem, np.:
 - `cp abc.txt xyz.txt` – kopiuje plik `abc.txt` pod nową nazwą `xyz.txt` w katalogu bieżącym;
 - `cp /tmp/abc.txt ~` – kopiuje plik `abc.txt` z katalogu `/tmp` do katalogu domowego użytkownika;
 - `cp abc.txt ~/xyz.txt` – kopiuje plik `abc.txt` z katalogu bieżącego pod nową nazwą `xyz.txt` w katalogu domowym użytkownika.
- Przydatnym przełącznikiem polecenia `cp` jest przełącznik `-r`, który służy do kopiowania całych struktur katalogów.

- **rm** [*przełączniki*] *lista_plików* – usuwanie plików podanych jako argumenty wywołania, np.:
rm abc.txt xyz.txt – usuwa pliki **abc.txt** i **xyz.txt** w katalogu bieżącym;
rm /tmp/abc.txt – usuwa plik **abc.txt** z katalogu **/tmp**;
Przydatnym przełącznikiem polecenia **rm** jest przełącznik **-r**, który służy do usuwania całych struktur katalogów.
- **mv** [*przełączniki*] *nazwa_pliku nowa_nazwa* – zmiana nazwy pliku określonego pierwszym argumentem wywołania na nazwę określoną drugim argumentem wywołania. Jeśli drugi argument wywołania jest katalogiem, to wówczas plik zostanie przeniesiony do tego katalogu, np.:
mv abc.txt xyz.txt – zmiana nazwy pliku **abc.txt** na nazwę **xyz.txt** w katalogu bieżącym;
mv /tmp/abc.txt ~ – przeniesienie pliku **abc.txt** z katalogu **/tmp** do katalogu domowego użytkownika.
- **touch** [*przełączniki*] *nazwa_pliku* – modyfikuje informacje na temat czasów modyfikacji i odczytu pliku, ale pozwala także na utworzenie pliku, np.:
touch abc.txt – utworzenie (pustego) pliku **abc.txt** w katalogu bieżącym.
Polecenia dotyczące plików (i katalogów) można także wydawać z wykorzystaniem tzw. wzorców uogólniających, które tworzy się z zastosowaniem następujących operatorów:
 - ***** – zastępuje dowolny ciąg znaków (także pusty);
 - **?** – zastępuje dokładnie jeden dowolny znak;
 - [**<znaki>**] – zastępuje dokładnie jeden znak z podanego zakresu, np.: [**xyz**];
 - [**^<znaki>**] – znak **^** na początku oznacza dopełnienie zbioru, czyli dla przykładu [**^xyz**], oznacza dowolny znak nie będący literą **x**, **y** i **z**.
Oto przykładowe polecenia z wykorzystaniem wzorców uogólniających:
 - **cp ./*.txt ~** – kopiowanie wszystkich plików z rozszerzeniem **.txt** z katalogu bieżącego do katalogu domowego użytkownika;
 - **rm ./[0-9]*** – usunięcie wszystkich plików z katalogu domowego, których nazwa rozpoczyna się od cyfry.

VII. Wyszukiwanie plików.

Jak już wspomniano pliki w systemach UNIX są używane do przechowywania danych użytkowników oraz reprezentują m. in. niektóre urządzenia systemowe, istotne jest zatem sprawne wyszukiwanie i lokalizowanie plików w strukturze katalogów. Zadanie to można zrealizować na kilka sposobów, w zależności od charakteru poszukiwanego pliku i kryteriów wyszukiwania.

Lokalizacji plików wykonywalnych – programów – można dokonać z wykorzystaniem polecenia:

```
whereis [przełączniki] lista_programów
```

Wynikiem działania tego polecenia jest informacja o pełnej ścieżce do pliku oraz lokalizacja pliku z pomocą systemową dla wyszukiwanego programu – przykład:

```
whereis ls  
ls: /bin/ls /usr/share/man/man1/ls.1.gz
```

Do wyszukiwania plików można także zastosować polecenie:

```
locate [przełączniki] wzorzec
```

Program ten wyszukuje pliki, podając listę plików ze ścieżkami, których nazwa zostanie dopasowana do podanego jako argument wywołania wzorca. Program **locate** zwraca wyniki niemal natychmiast, ponieważ wyszukiwanie faktycznie odbywa się na przygotowanej wcześniej bazie plików (nie jest przeszukiwana cała struktura katalogów). Baza ta (indeks nazw plików), przeważnie jest aktualizowana raz na dobę – oznacza to, że wynik może nie uwzględniać zmian w systemie plików i katalogów, jakie zostały wykonane po ostatniej aktualizacji spisu (aktualizacji może zawsze dokonać administrator systemu wykorzystując polecenie **updatedb**). Wzorzec zapytania można

budować z wykorzystaniem operatorów uogólniających – wówczas należy umieścić wzorzec w cudzysłowie, aby nie został on rozwinięty przez interpreter poleceń; oto przykłady wywołań polecenia `locate`:

```
locate ls
locate "*"s"
locate "/usr/share/man/*ls.1.gz"
```

Przykład zastosowania polecenia `locate` prezentuje Rysunek 4.

```
[adam@unix temp]$ locate "/usr/share/man/*ls.1.gz"
/usr/share/man/man1/fluidballs.1.gz
/usr/share/man/man1/ls.1.gz
/usr/share/man/man1/metaballs.1.gz
/usr/share/man/man1/mtools.1.gz
/usr/share/man/man1/nsgmls.1.gz
/usr/share/man/man1/onsgmls.1.gz
/usr/share/man/man1/smbcacls.1.gz
/usr/share/man/pl/man1/fileutils.1.gz
/usr/share/man/pl/man1/ls.1.gz
```

Rysunek 4: Przykładowy wynik działania polecenia `locate`

Faktyczne przeszukiwanie struktury katalogów systemu w poszukiwaniu plików (i katalogów) można zrealizować stosując polecenie:

find *katalog kryteria*

Pierwszym argumentem wywołania polecenia `find` jest nazwa katalogu, od którego ma się rozpocząć poszukiwanie; drugi argument dotyczy kryteriów jakie mają dotyczyć wyszukiwania. Specyfikacja kryteriów wyszukiwania to koniunkcja kryteriów, z których najczęściej używane to:

- **-name** <nazwa> – pozycje o podanej nazwie (można także stosować operatory uogólniające) – rozróżniana jest jednak wielkość liter;
- **-iname** <nazwa> – pozycje o podanej nazwie (można także stosować operatory uogólniające) – brak rozróżniania wielkości liter;
- **-size** <rozmiar><jednostka> – pozycje o określonym rozmiarze – możliwe są następujące określenia jednostek: **c** – bajty, **k** – kilobajty, **w** – dwubajtowe słowa, brak jednostki oznacza 512-bajtowe bloki; przed rozmiarem można podać znaki **+** lub **-**, oznaczają one wówczas odpowiednio: pozycje o rozmiarze większym/mniejszym niż podany;
- **-atime** <dni> – pozycje, na których była wykonywana jakaś operacja podaną liczbę dni temu; przed ilością dni można podać znaki **+** lub **-**, oznaczają one wówczas odpowiednio: pozycje, na których była wykonywana jakaś operacja więcej/mniej dni temu;
- **-mtime** <dni> – pozycje, na które były modyfikowane podaną liczbę dni temu; przed ilością dni można podać znaki **+** lub **-**, oznaczają one wówczas odpowiednio: pozycje, na które były modyfikowane więcej/mniej dni temu;
- **-ctime** <dni> – pozycje, których *i-węzeł* był modyfikowany podaną liczbę dni temu; przed ilością dni można podać znaki **+** lub **-**, oznaczają one wówczas odpowiednio: pozycje, których *i-węzeł* był modyfikowany więcej/mniej dni temu;
- **-type** <typ> – określenie jakie pozycje mają zostać odnalezione: **f** – pliki zwykłe, **d** – katalogi, **l** – *dowiązania symboliczne*, **c** – *urządzenia znakowe* (niebuforowane), **b** – *urządzenia blokowe* (buforowane), **p** – *kolejki FIFO*;
- **-exec** <polecenie> [{}] \; – wykonuje dowolne polecenie; polecenie może zostać wykonane na odszukanych pozycjach – wówczas należy zastosować jako argument polecenia znaki {}.

Oto przykłady użycia polecenia **find**:

- **find ~ -name abc.txt** – wyszuka wszystkie pozycje o nazwie **abc.txt**, które znajdują się w katalogu domowym użytkownika (oraz podkatalogach);
- **find ~/temp -name "*.txt"** – wyszuka wszystkie pozycje o nazwie z rozszerzeniem **.txt**, które znajdują się w katalogu **temp** (oraz jego ewentualnych podkatalogach) w katalogu domowym użytkownika ;
- **find ~ -iname "*.txt" -type f -size +100k** – wyszuka wszystkie pliki zwykłe w katalogu domowym użytkownika (i jego podkatalogach), które mają rozszerzenie **.txt** – wielkość liter bez znaczenia – oraz rozmiar większy niż 100kB;
- **find /tmp -type f -atime +2 -exec rm {} \; -print** – wyszuka wszystkie pliki zwykłe w katalogu **/tmp** (oraz jego podkatalogach), na których nie były wykonywane żadne operacje w ciągu ostatnich 48 godzin oraz usunie wszystkie odnalezione pliki; dodatkowy przełącznik **-print** powoduje, że zostaną wyświetlone nazwy odnalezionych plików, pomimo wykonania na nich dodatkowej operacji (tutaj **rm**).

VIII. Prawa dostępu.

W systemach UNIX dostęp do plików i katalogów zabezpieczony jest tzw. *prawami dostępu*, które regulują zasady na jakich użytkownicy mogą korzystać z tych zasobów. Wyróżnia się trzy rodzaje praw: prawo odczytu – oznaczane *r* (ang. *read*), prawo zapisu – oznaczane *w* (ang. *write*) oraz prawo wykonania – oznaczane *x* (ang. *execute*). Takie prawa są określane niezależnie dla: użytkownika, który jest właścicielem pliku lub katalogu (domyślnie właścicielem jest użytkownik, który utworzył dany plik lub katalog); użytkowników, którzy należą do tej samej grupy, do której należy plik lub katalog oraz dla pozostałych użytkowników. Interpretacja praw dostępu jest następująca (Tabela 1):

<i>Czynność do wykonania</i>	<i>Prawa do pliku</i>	<i>Prawa do katalogu</i>
Przeglądanie zawartości katalogu	---	r ---
Utworzenie pliku w katalogu	---	- w x
Zmiana nazwy pliku w katalogu	---	- w x
Usunięcie pliku z katalogu	---	- w x
Odczytanie zawartości pliku	r --	-- x
Zapis do pliku	- w -	-- x
Wykonanie pliku (np. programu lub skryptu)	-- x	-- x

Tabela 1: Interpretacja praw dostępu

Jak już wspomniano informacje o prawach dostępu można uzyskać dzięki poleceniu **ls** z przełącznikiem **-l** – oto przykład oraz jego interpretacja:

```
ls -l
drwx----- 15 adam students 4096 lip 6 13:27 ./
drwxr-xr-x 54 adam students 4096 lip 6 11:20 ../
-rwxr--r-x 2 adam students 4096 cze 23 13:32 abc.txt
```

Informacja o prawach wyświetlana jest według następującego schematu (Tabela 2):

<i>Użytkownik (user)</i>			<i>Grupa (group)</i>			<i>Inni użytkownicy (others)</i>		
r	w	x	r	w	x	r	w	x

Tabela 2: Prawa dostępu

Zatem dla pliku **abc.txt** dostępne są następujące prawa (znak ”-“ oznacza brak danego prawa): dla właściciela dostępne są wszystkie prawa, dla członków grupy **students** dostępne jest tylko prawo do odczytu, a dla pozostałych użytkowników prawa odczytu i wykonywania.

Prawami dostępu można także operować stosując notację numeryczną, w której każde prawo ma przypisaną pewną wartość liczbową, i tak: prawo odczytu – 4; prawo zapisu – 2, prawo wykonywania – 1. Tak więc, prawa zapisane numerycznie dla pliku **abc.txt** z powyższego przykładu miałyby następującą postać:

745 – 7 oznacza wszystkie prawa dla użytkownika (4 + 3 + 1), 4 oznacza prawo odczytu dla grupy, a 5 oznacza praw odczytu i wykonywania (4 + 1) dla pozostałych użytkowników.

Operowanie prawami dostępu i określaniem prawa własności jest możliwe dzięki następującym poleceniom systemowym:

- **chmod** [przełączniki] *uprawnienia nazwa_pliku_lub_katalogu* – zmiana praw dostępu wskazanych pierwszym argumentem wywołania dla pliku lub katalogu wskazanym drugim argumentem wywołania; możliwości wywołania tego polecenia przedstawia Rysunek 5:

chmod	u g o a	+ - =	r w x	<i>lista_plików</i>
--------------	------------------	-------------	-------------	---------------------

Rysunek 5: Składania polecenia **chmod**

W specyfikacji należy zatem wskazać (i) dla kogo mają być zmienione prawa (u – właściciel, g – użytkownicy z tej samej grupy, o – inni użytkownicy, a – wszyscy), (ii) rodzaj zmiany (+ – dodanie praw, - – odjęcie praw, = – ustalenie praw) oraz (iii) prawa. Oto przykładowe zlecenia z wykorzystaniem polecenia **chmod**:

chmod u+w plik.txt – dodaje prawo odczytu dla właściciela do pliku **plik.txt**;

chmod go-x plik.txt – usuwa prawo wykonywania dla użytkowników z tej samej grupy i innych do pliku **plik.txt**;

chmod a=r plik.txt – ustawia prawa dostępu na tylko do odczytu dla wszystkich użytkowników do pliku **plik.txt**;

Polecenie **chmod** umożliwia także określanie praw dostępu w postaci numerycznej, np.:

chmod 777 plik.txt – ustawia wszystkie prawa, wszystkim użytkownikom do pliku **plik.txt**;

chmod 742 – ustawia prawa odczytu, zapisu i wykonywania właścicielowi, prawo odczytu użytkownikom z tej samej grupy oraz prawo zapisu innym użytkownikom do pliku **plik.txt**;

- **chown** [przełączniki] *nazwa_nowego_właściciela nazwa_pliku_lub_katalogu* – zmiana właściciela pliku lub katalogu. Ze względu na nieodwracalność ewentualnych zmian, polecenie to jest często zarezerwowane dla administratora systemu.
- **chgrp** [przełączniki] *nazwa_nowej_grupy nazwa_pliku_lub_katalogu* – zmienia grupę, do której należy wskazany plik lub katalog. Podobnie jak polecenie **chown**, i to polecenie najczęściej jest zarezerwowane dla administratora.

IX. Dowiązania.

W systemach UNIX informacje o plikach na dysku przechowywane są w strukturach, które nazywa się *i-węzłami* (ang. *i-node*) – każdy taki *i-węzeł* przechowuje m.in. następujące informacje: prawa dostępu, daty ostatnich modyfikacji, *licznik dowiązań* oraz dodatkowe atrybuty. Licznik dowiązań określa ile razy dany plik dostępny jest w systemie plików – być może w różnych katalog pod różnymi nazwami. Licznik ten umożliwia realizację dowiązań do plików, które z kolei są dodatkowymi nazwami dla pliku, umożliwiającymi dostęp do oryginału (np. z poziomu różnych

katalogów).

Istnieją dwa rodzaje dowiązań: tzw. dowiązania twarde (ang. *hard links*) oraz tzw. dowiązania miękkie lub symboliczne (ang. *soft or symbolic links*). Dowiązania symboliczne mogą także dotyczyć katalogów, oraz plików w innych systemach plików – informacja o nich dostępna jest dzięki omówionemu już poleceniu `ls -l`. Wszystkie dowiązania można przetwarzać dokładnie tak samo jak pliki zwykle, w szczególności mogą także być usunięte poleceniem `rm`.

Tworzenie dowiązań jest możliwe dzięki poleceniu:

```
ln [przełączniki] źródło nazwa_dowiązania
```

pierwszy argument musi wskazywać na istniejący plik (lub katalog w przypadku dowiązań symbolicznych), do którego tworzone jest dowiązanie, a drugim argumentem jest nowa nazwa dla tego pliku. Utworzenie dowiązania symbolicznego wymaga zastosowania przełącznika `-s`. Przykładowe wywołania zlecenia utworzenia dowiązań:

- `ln ./abc/plik.txt plik1.txt` – tworzy dowiązanie (twarde) do pliku `plik.txt` w katalogu `./abc` pod nazwą `plik1.txt` w katalogu bieżącym;
- `ln -s ./abc/plik.txt ~/plik1.txt` – tworzy dowiązanie symboliczne do pliku `plik.txt` w katalogu `./abc` pod nazwą `plik1.txt` w katalogu domowym użytkownika.

X. Zadania do samodzielnego wykonania.

- 1) Zaloguj się do systemu.
- 2) Zmień własne hasło.
- 3) Sprawdź własny identyfikator oraz grupy, do których należysz.
- 4) Sprawdź kto jest zalogowany w chwili obecnej w systemie.
- 5) Zapoznaj się z opisem struktury katalogów – polecenie `man 7 hier`.
- 6) Wyświetl zawartość katalogu domowego.
- 7) Wyświetl zawartość podstawowych katalogów w systemie (np. `/dev`, `/etc`, `/home`, `/usr`).
- 8) Utwórz katalog `kat1` w katalogu domowym.
- 9) W katalogu `kat1` utwórz jednym poleceniem strukturę katalogów `kat2/kat3/kat4`.
- 10) Usuń jednym poleceniem całą strukturę katalogów `kat3/kat4`.
- 11) Utwórz w katalogu domowym pliki o dowolnych nazwach z rozszerzeniami `.txt` i `.c`.
- 12) Skopiuj jednym poleceniem wszystkie pliki z katalogu domowego z rozszerzeniem `.txt` do katalogu `kat1`.
- 13) Skopiuj jednym poleceniem wszystkie pliki z katalogu domowego z rozszerzeniem `.c` do katalogu `kat2`.
- 14) Skopiuj całą strukturę katalogów `kat1` tworząc analogiczną strukturę o nazwie `kat1b`.
- 15) Usuń wszystkie pliki z katalogu `kat1/kat2`.
- 16) Usuń jednym poleceniem całą strukturę katalogów `kat1b`.
- 17) Zmień nazwę dowolnego pliku w katalogu `kat1`.
- 18) Przenieś katalog `kat1/kat2` do katalogu domowego tworząc w ten sposób katalog `kat2b`.
- 19) Przy pomocy programu `locate` znajdź wszystkie pozycje, które posiadają w nazwie słowo `mozilla`.
- 20) Przy pomocy programu `locate` znajdź wszystkie pozycje, które posiadają w nazwie słowo `mozilla` i znajdują się w podkatalogach katalogu `/usr`.
- 21) Korzystając z programu `find` znajdź wszystkie pliki, które posiadają w nazwie słowo `mozilla` i znajdują się w podkatalogach katalogu `/usr`.
- 22) Korzystając z programu `find` znajdź wszystkie katalogi o nazwie `bin`, które znajdują się w katalogu `/usr`.
- 23) Skopiuj wszystkie pliki zwykle o rozmiarze pomiędzy 10 a 100 bajtów z katalogu `/usr/bin` do katalogu `kat1/kat2` (wykorzystaj polecenie `find` z parametrem `-exec`).
- 24) W katalogu domowym utwórz plik o nazwie `plik.txt` – sprawdź jakie są prawa dostępu do

niego.

- 25) Dla pliku `plik.txt` dodaj prawo zapisu dla grupy.
- 26) Dla pliku `plik.txt` odejmij prawo zapisu dla właściciela.
- 27) Dla pliku `plik.txt` dodaj prawo wykonywania dla wszystkich użytkowników.
- 28) Dla pliku `plik.txt` przywróć oryginalne prawa korzystając z notacji numerycznej.
- 29) Utwórz dowiązanie do pliku `plik.txt` o nazwie `plik2.txt` w katalogu domowym.
- 30) Utwórz dowiązanie symboliczne do katalogu `kat1/kat2` o nazwie `abc` w katalogu domowym.
- 31) Przejrzyj pomoc systemową dla wszystkich poleceń zaprezentowanych podczas zajęć.

XI. Literatura.

- [Bac95] Bach M. J., *Budowa Systemu Operacyjnego UNIX*, WNT, 1995, ISBN 83-204-2015-6.
- [Kof00] Kofler M., *Linux. Instalowanie, konfigurowanie, użytkowanie*, Wydawnictwo Translator, ISBN 83-86149-63-9.
- [Sob01] Sobaniec C., *Linux – Przewodnik Użytkownika.*, Wydawnictwo NAKOM, 2001, ISBN 83-86969-53-9.