Treść:

Przygotować: Podsystem plików –poziom fizyczny, istniejące systemy plików (FAT, NTFS, ext2,ext3, ext4, NFS, itd.) – sposoby realizacji poziomu fizycznego, pojęcia: partycja, jednostka alokacji, ścieżka dostępu.
Na zajęciach: Podsystem plików –poziom fizyczny, powiązanie poziomu fizycznego z logicznym, istniejące systemy plików (FAT, NTFS, ext2,ext3,..., ext5, NFS, itd.) – sposoby realizacji poziomu fizycznego, pojęcia: partycja, jednostki alokacji, tablica alokacji, ścieżka dostępu.
Ćwiczenia: zarządzanie zasobami użytkownika w ramach SO Linux – kontrola dostępu.

Systemy pomocy w Linuxie – man, info.

Linux ma wbudowany podręcznik systemowy, zawierający 8 rozdziałów:
1- Polecenia użytkownika
2- Niskopoziomowe wywołania systemowe
3- Dokumentacja wysokopoziomowych bibliotek Uniksa
4- Informacja o interfejsach urządzeń i sterownikach
5- Opisy plików (konfiguracji systemu)
6- Gry
7- Formaty plików, konwencje i kodowania (ASCII, przyrostki itd.)
8- Polecenia systemowe i serwery

Dostęp do zawartości można uzyskać poprzez polecenie:
man <polecenie> np.: *man ls*
man –k <słowo_kluczowe> np.: *man –k sort*
Otrzymana informacja:
comm(1) – compare ... <= w nawiasach nr rozdziału
qsort(3) – sorts an array
...
Można się odwołać do polecenia z danego rozdziału, np.:
man 5 passwd - opis pliku passwd w rozdziale 5-tym.
WYJŚCIE – „q”.

Jest również system pomocy GNU – pliki dokumentacji.
Dostęp do tych zasobów:
info <polecenie>
WYJŚCIE – „q”.
Autorytatywnym źródłem jest obecnie dokumentacja Texinfo. Dostęp do niej uzyskasz wpisując w wierszu poleceń:
pinfo ls
lub

info ls.

Ta strona podręcznika man opisuje polecenie ls w wersji GNU.

Filtry stosowane w poleceniach (tzw. konwencja *):

Np. cp *.txt /Dane

cp ???.txt /Dane

cp *a* /Dane

* - zastępuje dowolną ilość znaków

? - zastępuje dokładnie jeden znak

[listaZnakow] - jeden ze znaków z listy na tej pozycji

np. - [a-z] dowolna mała litera

ls ??[a-k,p,t]* - pokaż zasoby o nazwie zawierającej na trzeciej pozycji jeden ze znaków: a-k,p,t, nazwa powinna zawierać przynajmniej trzy znaki.

Kontrola dostępu do zasobów:

W Linux-ie przyjęto zasadę gradacji dostępu - trzy poziomy - właściciel, grupa lub grupy (stowarzyszona), reszta.

W kontekście każdego z poziomów ustalane są prawa do wykonywanych operacji:

r - read - odczyt

w - write - zapis

x - execute - wykonanie.

Dają one uprawnienia do:

- r - odczytu pliku, czyli przeglądania jego zawartości. W przypadku katalogów będzie to oznaczać możliwość wypisania listy plików i podkatalogów w nim umieszczonych,
- w - zapisu pliku, czyli modyfikowanie jego zawartości. W przypadku katalogów możliwe jest tworzenie i usuwanie plików w tym katalogu,
- x - uruchomienia pliku wykonywalnego (programu, lub skryptu). W przypadku katalogów oznacza to dostęp do listy plików w nim zapisanych, oraz możliwość przejścia do tego katalogu, np. przez polecenie cd katalog.

Tą informację pokazuje np. polecenie *ls -l*

Konwencja pokazywania w *ls -l*:

jest znaczek= jest prawo

jest '-' = brak prawa.

ls -l <= inf. o plikach:

- rw-rw-r-- 1 st_2014 st_2014 0 mar 1 14:18 Lista.txt

drwxr-xr-x 2 ... .. Dokumenty

INF. od Podsystemu zarządzania plikami - o dostępie do zasobu.

-rw-rw-r--

Pierwszy znak - tu '-' - oznacza typ pliku.

OBECNIE W SO - jest zasada - wszystko jest plikiem (każdy zasób traktowany jest jak plik) - jako plik możemy potraktować: folder, klawiaturę, monitor, myszkę, partycję, plik, katalog, socket (gniazdo), (urządzenia blokowe, urządzenia znakowe), ...

Np.:

„-" - oznacza plik zwykły

d - oznacza plik typu directory = katalog,
inne: l (link – dowiązanie symboliczne), b (block – urządzenie blokowe), c (char – urządzenie znakowe), p (pipe – łącze nazwane), s (socket - gniazdo), ...
Następne - 3 x 3 - opis praw dostępu do zasobu.
Pytanie - które prawo pozwala (tak naprawdę) na usunięcie pliku?

Polecenie :
chmod <nowe prawa> <nazwa pliku>
służy do zmiany praw dostępu.
Nowe prawa - możemy zapisać za pomocą notacji 777:
111 111 111
rwx rwx rwx
101 100 110
r -x r - - rw-
5 4 6
chmod 546 plik1
1=właściciel -5
2=grupa - 4
3= reszta - 6.
Inny sposób zapisu polecenia
-rwx r- x r- - ... plik
111101100 => 754 *chmod 754 plik1* .
Chcemy dodać prawo w dla grupy, usunąć prawo r dla reszty:
chmod g+w,o-r plik
u- user, g –group, o –other .
Można też przypisywać konkretne prawa niezależnie od tego co było wcześniej:
chmod u=rwx, g=rw,o-r plik .

Każdy zasób w Linuxie ma swego właściciela.
Można zmienić właściciela zasobu za pomocą polecenia:
chown [opcje] nazwa_właściciela nazwa_zasobu,
może to wykonać użytkownik uprawniony – właściciel lub root.
np.:
chown jan23 Zapiski.txt .
Można z opcją –R zmienić właściciela dla całego katalogu z podkatalogami (R – rekursywnie) –np.- jeżeli Dane jest katalogiem z podkatalogami i plikami to zmiana właściciela dla całości:
chown –R jan23 Dane

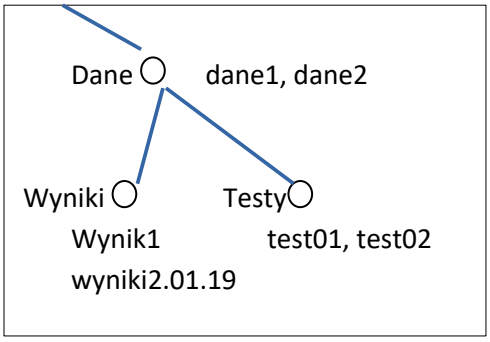
W środowisku bash-a jest dla każdego użytkownika zdefiniowana tzw. maska określająca jakie prawa dostępu będą nadawane domyślnie dla nowo tworzonych zasobów. Jej wartość jest zapisana w specjalnym pliku danego użytkownika i odczytywana w momencie jego logowania do systemu. Działa to tak, że suma maski i nadawanych praw daje:
-0666 dla plików
-0777 dla katalogów.
Sprawdzenie – jaka maska jest ustawiona – polecenie:

umask
0002
(pierwsza cyfra dotyczy uprawnień, o których będzie mowa później – m.in. tzw. bitu lepkości).
Bierzemy pod uwagę trzy pozostałe cyfry:
-dla plików: 0666-0002 =0664 – czyli: rw-rw-r—
-dla katalogów: 0777-0002 = 0775 – czyli: rwxrwxr-x.
Możemy na czas bieżącego logowania zmienić maskę – to samo polecenie:
umask 0077

Poziom logiczny a poziom fizyczny:

A) Co tak naprawdę powinno się kojarzyć z katalogiem, katalog jako plik, zawartość katalogu?

Katalog – specjalny plik – do obsługi konstrukcji podkatalogów. Jeżeli to jest plik – to co jest w nim zapisane? Zawartość katalogu – pliki i podkatalogi które w nim się znajdują. Katalog – to plik.
Przykład: podkatalog „Dane”:



Zawartość katalogu Dane: podkatalogi Wyniki, Testy; pliki: dane1, dane2.

W systemie ext2 - co znajduje się w pliku tego katalogu w pewnym uproszczeniu – zawiera tabela.
Dla ścieżki. **/Dane/Wyniki przejścia** do poziomu fizycznego opisane będą w tabelach:

Tabela 1 – zawartość kat. bieżącego

Nazwa	**Nr i-węzła**
Dane	655
...	

Tabela 2 – zawartość kat. Dane

Nazwa	**Nr i-węzła**
Wyniki	822
Testy	855
dane1	1010
dane2	1013

Tabela 3 – zawartość kat. Wyniki

Nazwa	**Nr i-węzła**
Wynik1	1073
wyniki2.01.19	1233

Tabela pozwala na przejście z poziomu logicznego systemu plików na poziom fizyczny. Lewa kolumna – poziom logiczny – nazwa zasobu, prawa – numer odpowiadającego zasobowi i-węzła w tablicy i-węzłów – poziom fizyczny.

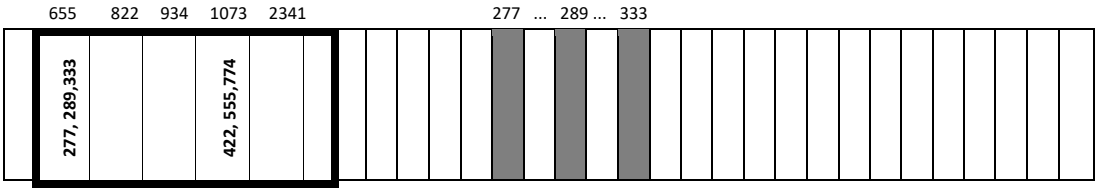


Tabela i-węzłów

bloki

W i-węźle zapisane są m. in. numery bloków, w których znajduje się treść pliku.
Jeżeli dla pliku opisującego zawartość katalogu Dane przypisany jest i-węzeł o numerze 55, to w nim są zapisane numery bloków, gdzie zapisana jest „treść" tego pliku. Jeżeli te bloki to: 277, 289,333 – to odczytując ich zawartość – na poziomie logicznym otrzymamy Tabelę 1.
Aby odczytać np. zawartość pliku *dane1* – należy w Tabeli 1 znaleźć numer i-węzła mu odpowiadający - 1073, w i-węźle znaleźć numery bloków zawierające treść pliku - 422, 555,774, następnie odczytać zawartość tych bloków.

W systemie FAT16 (32) sytuacja będzie wyglądała podobnie – ale oczywiście z niuansami.
W pliku utożsamianym z tym katalogiem w pewnym uproszczeniu – będzie taka zawartość – Tabela 2:

Tabela 2

Nazwa	**Nr 1-go klastra pliku**
Wyniki	122
Testy	234
dane1	277
dane2	2341

Tabela pozwala na przejście z poziomu logicznego systemu plików na poziom fizyczny. Lewa kolumna – poziom logiczny – nazwa zasobu, prawa – numer zapisu w tabeli FAT dotyczącego danego pliku = numer pierwszego klastra zawierającego dane pliku - poziom fizyczny. Sytuację w partycji – dla pliku dane1 – pierwszy klaster ma nr 277, treść pliku jest w klastrach: 277,289,333 – przedstawia poniższy schemat:

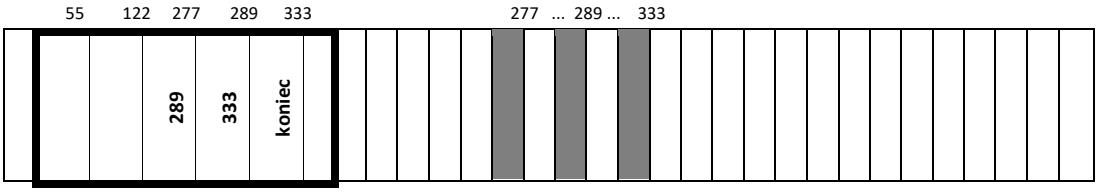
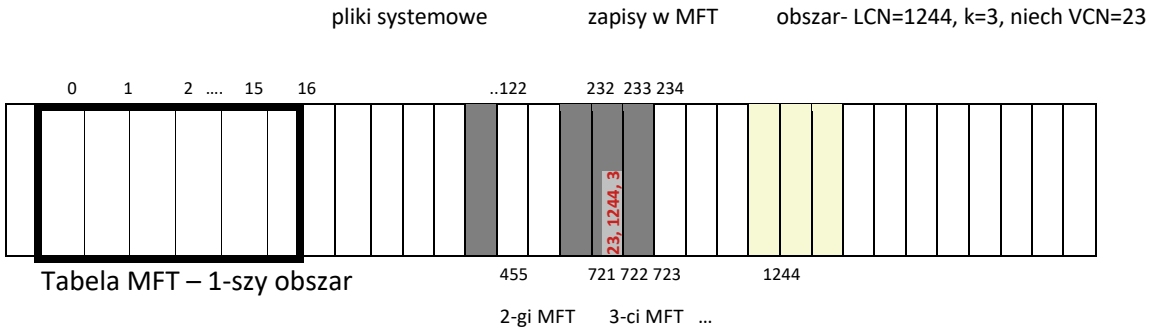


Tabela FAT

klastry

W pojedynczym zapisie w tabeli FAT jest m. in. podany numer kolejnego klastra, w którym znajduje się treść pliku. Zapis dotyczący ostatniego klastra pliku zawiera specjalną informację – o tym, że to jest koniec pliku.
Jeżeli klaster nie jest zajęty – w tabeli FAT jest stosowny zapis.
Wniosek – tabela FAT powinna zawierać tyle zapisów – ile jest klastrów w partycji.

W systemie NTFS struktura partycji jest bardziej skomplikowana. Mamy partycje. W ramach partycji mamy tabelę rozmieszczenia plików oraz obszar danych. Partycja dzielona jest na klastry. Klastry w partycji są ponumerowane. Tabela partycji – MFT – Master File Table.
Tabela MFT ma swoją część zasadniczą, która mieści się na początku partycji, pozostałe jej części mogą być zapisywane naprzemiennie z obszarami danych.
Obszar danych składa się z szeregu klastrów.
Obszar danych adresowany jest poprzez adres składający się z trzech części: VCN, LCN, k,
gdzie:
VCN – numer pierwszego klastra obszaru w ramach pliku
LCN – numer pierwszego klastra obszaru w ramach partycji
k – ilość klastrów w obszarze.



Zawartość katalogu opisana jest za pomocą tabeli – jak w ext2 i FAT (Tabela1, Tabela 2). W tabeli opisującej zawartość katalogu podaje się numer zapisu przyporządkowanego dla danego zasobu w tabeli MFT.

Tabela 3

Nazwa	**Nr zapisu w MFT**
Wyniki	122
Testy	134
dane1	233
dane2	2341

Np. dla pliku dane1 – nr zapisu w MFT – 233. Jak widać z obrazka – plik jest z kategorii duży – jego dane mieszczą się w jednym zapisie w MFT i w dodatkowym obszarze o rozmiarze 3 klastrów.

Ale sytuacja się komplikuje – gdyż w systemie rozróżnia się ze względu na rozmiar – 4 wielkości plików: małe (small), duże (large), bardzo duże (huge), ekstremalnie duże (extremely huge).

Struktura jednego zapisu w tabeli MFT – mały plik:

SI	FN	Data	SD

SI – standardowa informacja, FN – nazwa pliku, Data – dane pliku, SD – deskryptor bezpieczeństwa.

Małe mieszczą się w jednym zapisie w tabeli MFT, duże – jeden zapis w MFT i kilka obszarów (obszar zawiera zwarty ciąg klastrów) w przestrzeni danych, bardzo duże – mieszczą się w dwóch zapisach w MFT – pierwszy – ma strukturę jak w poprzednich typach plików, drugi jest używany do adresowania pośredniego - i pewnej ilości obszarów w przestrzeni danych, ekstremalnie duże – mają więcej zapisów w tabeli MFT do adresowania pośredniego.

B) Jak w oparciu o ścieżkę dostępu na poziomie logicznym znaleźć zawartość zasobu (pliku) na poziomie fizycznym?
Np. - *cat /home/WMII/s123456/dane.txt*
Odpowiedzi – należy sformułować w oparciu o informację zawartą w tekście powyżej.

C) Co to jest i-węzeł?
Polecenie *ls -i.*

Koncepcja pliku.

Koncepcja – „wszystko jest plikiem". Cel – unifikacja obsługi – typowe operacje na plikach.
Pliki nietypowe (drukarka, mysz, …) = plik specjalny. Operacje nie mające specjalnie sensu – zamienia się na atrapy – jest operacja, ale inaczej się wykonuje (np. zapis do myszy).
OBECNIE W SO - jest zasada - **wszystko jest plikiem** (każdy zasób traktowany jest jak plik) - jako plik możemy potraktować: folder, klawiaturę, monitor, myszkę, partycję, plik, katalog, socket (gniazdo), (urządzenia blokowe, urządzenia znakowe), ...
Np.:
- - oznacza plik zwykły
d - oznacza plik typu directory=katalog,
inne: l,b,c,p,s,... .

Pytanie – jak wygląda logiczna organizacja pliku – tzn. jakimi porcjami informacja jest w pliku zamieszczana – nie chodzi o rozmieszczenie na nośniku – tylko o to – czy plik to zestaw pewnych porcji o znanej strukturze czy też ciąg bitów?

Są dwa podejścia:
– 1-sze – dane z pliku są interpretowane wyłącznie przez aplikację – wtedy plik dla SO przedstawia sobą nieustrukturowany ciąg bitów – stosowane do zwykłych plików w Unixie, Linuxie, Windows, itp.,
- 2-gi – plik składa się z zapisów o zdefiniowanej budowie (strukturze) – logicznych rekordów - wtedy SO musi znać strukturę rekordów używanych w tego typu plikach i zapis/odczyt robić na poziomie tych rekordów. Do zwykłych plików stosowane rzadko, natomiast stosowane w SO do plików specjalnych – np. opisujących zawartość katalogu albo urządzeń specjalnych.

Przy pierwszym podejściu – zapis/odczyt sekwencyjny, można używać narzędzi pozwalających na przesunięcie wskaźnika do pliku o wybraną ilość bajtów od początku/końca pliku i ustalenie porcji danych do zapisu/odczytu.
Przy drugim – możliwa jest organizacja dostępu sekwencyjnego albo bezpośredniego do wybranego rekordu. Możliwa też jest indeksacja zapisów.

W Linux-ie – pliki związane z obsługą urządzeń zawarte są w podkatalogu /dev:

```
autofs           log                 ram0      tty1   tty36  tty62
bus              loop0               ram1      tty10  tty37  tty63
cdrom            loop1               ram10     tty11  tty38  tty7
cdrom-hdc        loop2               ram11     tty12  tty39  tty8
console          loop3               ram12     tty13  tty4   tty9
core             loop4               ram13     tty14  tty40  ttyS0
cpu              loop5               ram14     tty15  tty41  ttyS1
cpu_dma_latency  loop6               ram15     tty16  tty42  ttyS2
disk             loop7               ram2      tty17  tty43  ttyS3
dvd              MAKEDEV             ram3      tty18  tty44  urandom
dvd-hdc          mapper              ram4      tty19  tty45  usbdev1.1_ep00
fd               mcelog              ram5      tty2   tty46  usbdev1.1_ep81
full             md0                 ram6      tty20  tty47  usbdev1.2_ep00
gpmctl           mem                 ram7      tty21  tty48  usbdev1.2_ep81
hda              net                 ram8      tty22  tty49  vcs
hda1             network_latency     ram9      tty23  tty5   vcs2
hda2             network_throughput  ramdisk   tty24  tty50  vcs3
hda3             null                random    tty25  tty51  vcs4
hda4             nvram               rawctl    tty26  tty52  vcs5
hda5             oldmem              root      tty27  tty53  vcs6
hda6             parport0            rtc       tty28  tty54  vcsa
hda7             parport1            shm       tty29  tty55  vcsa2
hdc              parport2            snapshot  tty3   tty56  vcsa3
hidraw0          parport3            stderr    tty30  tty57  vcsa4
hpet             port                stdin     tty31  tty58  vcsa5
initctl          ppp                 stdout    tty32  tty59  vcsa6
input            ptmx                systty    tty33  tty6   X0R
js0              pts                 tty       tty34  tty60  zero
kmsg             ram                 tty0      tty35  tty61
```

Operacje na pliku – z poziomu użytkownika – wydają się oczywiste – zapisz do pliku, odczytaj z pliku. Natomiast SO musi wykonać wiele składowych działań.
W przypadku np. polecenia:
cat dane1 - odczytu zawartości pliku dane1
SO musi: wyszukać plik w strukturze katalogów, odczytać inf. o nim, sprawdzić prawa dostępu, jeżeli pozwalają – to: otworzyć plik w odpowiednim trybie – do odczytu, odczytać potrzebne dane, zamknąć plik.

Bufory.

Należy pamiętać o tym, że SO stosuje system podręcznej dyskowej pamięci *cache* - w pamięci operacyjnej RAM wydzielane są segmenty – bufory, w których przechowuje się ostatnio używane bloki plików – bo być może będą one jeszcze używane. Wtedy operacje na pliku faktycznie będą się odbywać zawartości buforów i nie będzie zgodności w zawartości buforów i plików na nośniku zewnętrznym. Co jakiś czas następuje zapis zawartości buforów do pamięci zewnętrznej – synchronizacja zawartości plików.

W Linuxie żeby używać jakiegoś nośnika (np. pendrive'a, CD, DVD, dysk zewnętrzny, itp.), który trzeba dołączyć do komputera – po jego dołączeniu trzeba go dołączyć do systemu plików – zamontować, przed wyjęciem - odmontować. W Windows montowanie dla pendrive'ów jest automatyczne (chociaż domontowywanie wskazane).

Przy domontowywaniu powinna nastąpić automatyczna synchronizacja zawartości plików, ale bezpieczniej jest wykonać polecenie

sync

które ją wymusi.

W przypadku awarii SO możliwa jest utrata danych zawartych w buforach – jeżeli nie nastąpiła synchronizacja.