

Systemy operacyjne

WYKŁAD 2

dr inż. Stanisława Plichta
splichta@ans-ns.edu.pl

Usługi systemu operacyjnego

- Wykonanie programu - system powinien móc załadować program do pamięci i rozpocząć jego wykonanie.
- Operacje WE/WY - wykonywany program może potrzebować operacji WE/WY odnoszących się do pliku lub jakiegoś urządzenia.
- Manipulowanie systemem plików - programy muszą zapisywać i odczytywać pliki, tworzyć i usuwać pliki przy użyciu ich nazw.
- Komunikacja - procesy wymagają wzajemnego kontaktu i wymiany informacji. Ma to miejsce albo poprzez pamięć dzieloną, albo poprzez przekazywanie komunikatów.
- Wykrywanie błędów - SO ciągle musi być powiadamiany o występowaniu błędów. Błędy mogą być różne – mogą pochodzić od urządzeń, albo pojawiać się w wykonywanym programie. Na wszystkie błędy SO musi reagować.

Usługi systemu operacyjnego przeznaczone do optymalizacji działania systemu

Odrębną grupę usług stanowi zbiór funkcji SO, które nie pomagają użytkownikowi, ale przeznaczone są do optymalizacji działania samego systemu

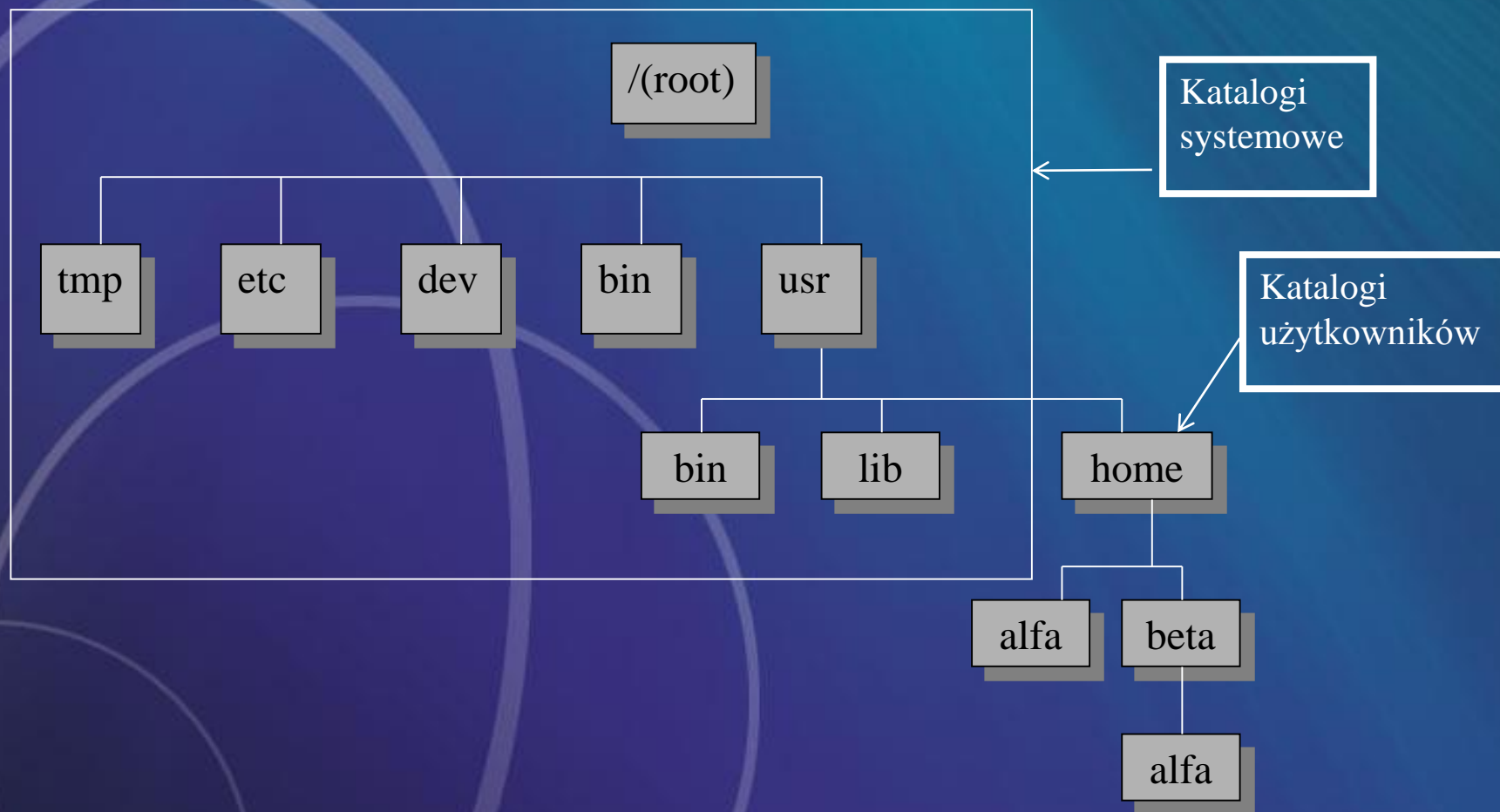
- Przydzielanie zasobów - procesora, pamięci, plików np. do określania najlepszego wykorzystania JC służą procedury planowania przydziału procesora.
- Rozliczanie - przechowywanie informacji o tym, którzy użytkownicy i w jakim stopniu korzystają z poszczególnych zasobów komputera.
- Ochrona - jeśli procesy wykonywane są współbieżnie, wówczas żaden proces nie powinien zaburzać pracy innych procesów lub samego SO. Ważne jest także zabezpieczenie systemu przed niepożądanymi czynnikami zewnętrznymi. Każdy użytkownik musi uwierzytelnić w systemie swoją tożsamość, na ogół za pomocą hasła.

Funkcje systemowe

Tworzą interfejs między wykonywanym programem
a systemem operacyjnym

- nadzorowanie procesów,
- operacje na plikach,
- operacje na urządzeniach,
- utrzymywanie informacji,
- komunikacja.

System Linux



Znaki specjalne nazw plików

ZNAKI SPECJALNE NAZW PLIKÓW: *, ?, []

- Gwiazdka * oznacza dowolny ciąg znaków, można jej używać do oznaczania nazw plików zaczynających się lub kończących dowolnym zestawem znaków
- Znak zapytania ? oznacza dokładnie jeden dowolny znak. Można używać więcej niż jednego znaku zapytania w każdym miejscu wzorca
- Dowolne znaki umieszczone w nawiasach kwadratowych [] oznaczają dokładnie jeden spośród wymienionych znaków

Znaki specjalne nazw plików

***.c** – pliki, które mają rozszerzenie

dok? – pliki, których nazwy rozpoczynają się od słowa „dok”, za którym następuje jeden dowolny znak,

dok[1x] – pliki, których nazwy rozpoczynają się od „dok”, a kończą się znakiem 1 lub x,

dok[A-D] – pliki, których nazwy rozpoczynają się od „dok”, a kończą się literą A, B, C lub D,

***** – wszystkie pliki z wyjątkiem ukrytych,

.* – wszystkie pliki ukryte.

Podstawowe polecenia

pwd – wyświetla bezwzględną ścieżkę do katalogu bieżącego użytkownika

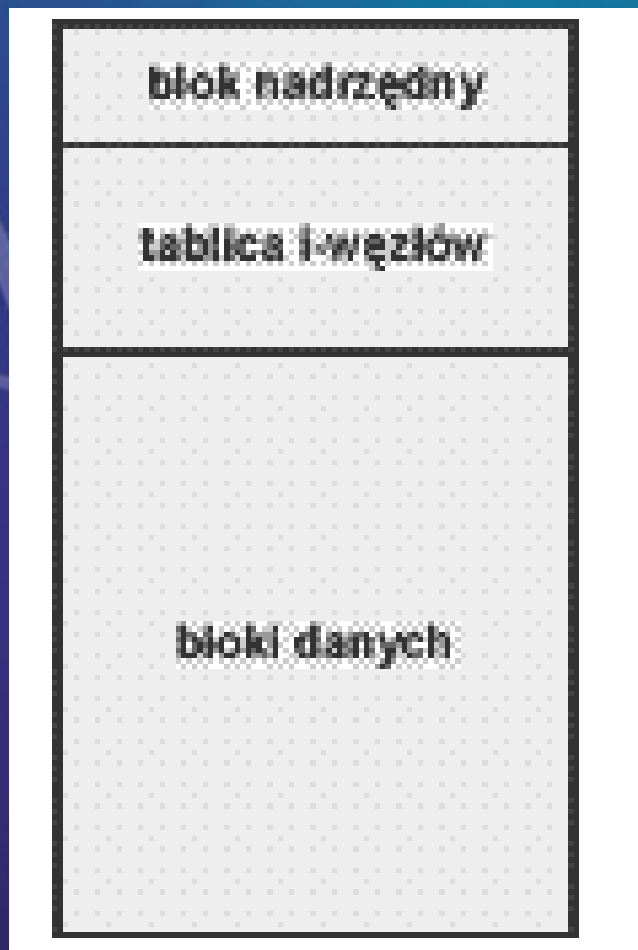
clear – czyści ekran, ustawia monit w lewym górnym rogu ekranu

who – wyświetla listę zalogowanych aktualnie użytkowników, ich terminali, godziny zalogowania oraz nazwę komputera, z którego się logowali

who am i – umożliwia wyświetlenie na ekranie danych o sobie

touch plik – tworzy pusty plik

Format patrycji - UNIX



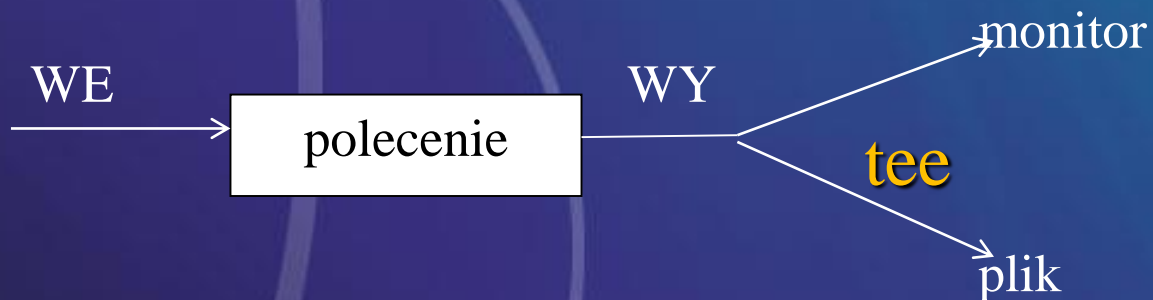
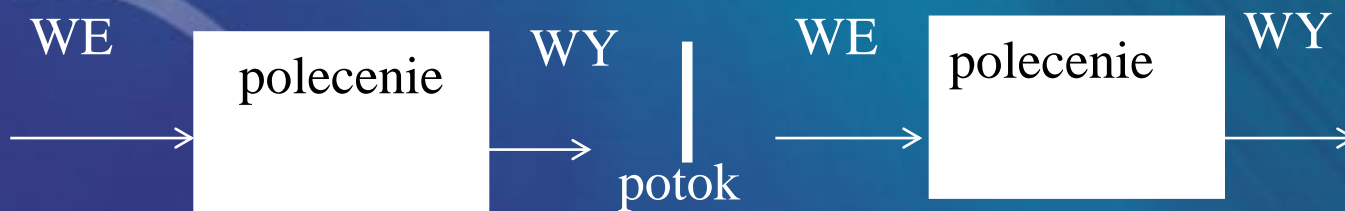
Struktura wpisu katalogowego



Struktura i-węzła

- identyfikator właściciela oraz grupy pliku
- typ pliku
- prawa dostępu
- rozmiar pliku w bajtach ostatni czas dostępu, modyfikacji
- czas utworzenia i skasowania
- liczba dowiązań
- liczba bloków dyskowych zajmowanych przez plik
- adresy dyskowe

Potoki



Operacje na katalogach

- Każdy katalog ma katalog nadrzędny (z wyjątkiem katalogu korzenia)
- Kiedy katalog jest tworzony, wykonywane są dwa wpisy:
 - kropka oznacza nazwę ścieżki katalogu bieżącego
 - .. dwie kropki – nazwę ścieżki katalogu nadrzędnego

Katalogi można zmieniać za pomocą polecenia **cd**

cd nazwa katalogu

Polecenie **cd** przyjmuje jako argument nazwę katalogu, do którego chcemy przejść

Operacje na katalogach

Katalogi można tworzyć poleceniem

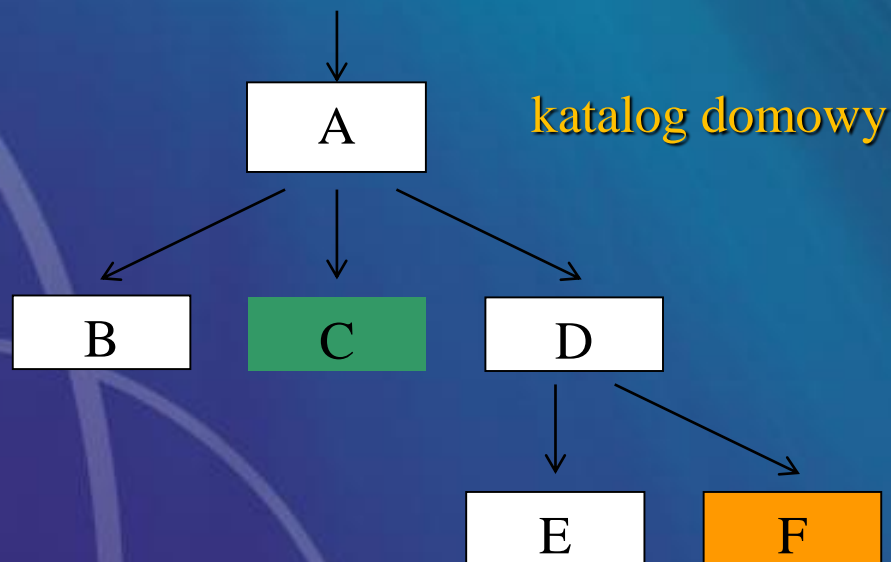
mkdir opcje katalogi

- p** utworzy zagnieżdżoną strukturę katalogów
- m** tworzy katalog z określonymi prawami dostępu

mkdir -p kat1/kat2

tworzy katalog kat1 z podkatalogiem kat2

Operacje na katalogach



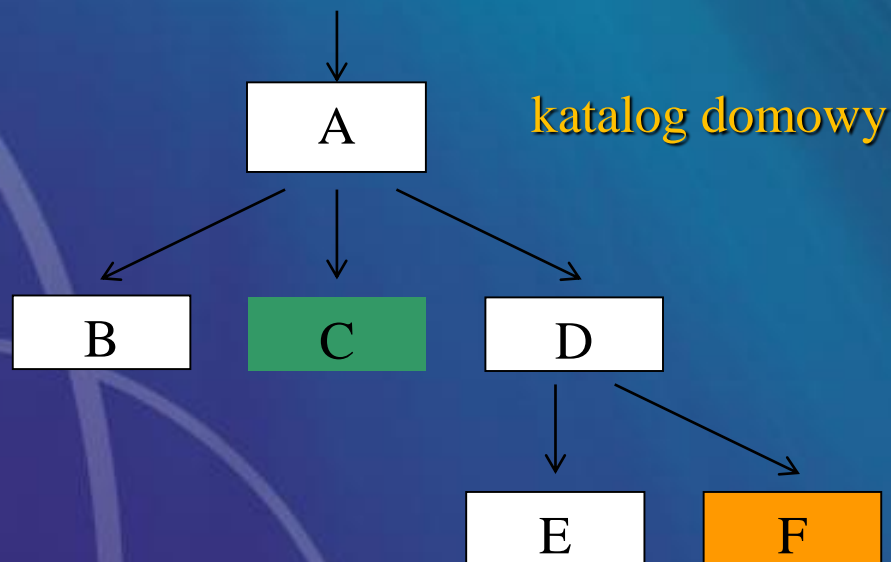
Ścieżka względna – przejście z katalogu F do katalogu C

```
cd ../../C
```

Ścieżka względna – przejście z katalogu C do katalogu F

```
cd ../D/F
```

Operacje na katalogach



Ścieżka bezwzględna – przejście z katalogu C do katalogu F

Ścieżka bezwzględna – przejście z katalogu F do katalogu C

`cd ~ /D/F`

`cd ~ /C`

Podstawowe polecenia

nazwa polecenia opcje argumenty

ls –opcje argumenty

- a**–wypisuje wszystkie pliki, razem z plikami ukrytymi,
- l**–podaje wszystkie dane (pełną informację) o plikach i katalogach,
- F**–dopisuje po każdym katalogu ukośnik, gwiazdkę po nazwach plików wykonywalnych i znak @ po nazwach plików powiązanych,
- R**–wypisuje wszystkie zagnieżdżone podkatalogi, umieszczone poniżej katalogu roboczego (rekurencyjne wyświetlanie podkatalogów wraz z zawartością),
- i**–wypisuje numer i-węzła dla każdego pliku,
- s**–podaje dodatkowo rozmiary pliku,

Podstawowe polecenia

nazwa polecenia opcje argumenty

ls –opcje argumenty

- S**—sortuje wg wielkości plików,
- t**—sortuje pliki wg daty ostatniej modyfikacji; nowsze pliki znajdują się na początku,
- u**—sortuje wg daty ostatniego dostępu,
- X**—wyświetla pliki posortowane według rozszerzeń.

ls -l *.c

ls -li ./

Podstawowe polecenia

```
stasia1@torus:~  
[stasia1@torus ~]$ ls -l  
razem 8  
drwx----- 2 stasia1 imk-prac 4096 pałś 16 12:00 kat1  
-rw----- 1 stasia1 imk-prac 25 pałś 16 12:00 plik1  
[stasia1@torus ~]$ ls -i  
4842870 kat1 4842927 plik1  
[stasia1@torus ~]$ ls -li  
razem 8  
4842870 drwx----- 2 stasia1 imk-prac 4096 pałś 16 12:00 kat1  
4842927 -rw----- 1 stasia1 imk-prac 25 pałś 16 12:00 plik1  
[stasia1@torus ~]$
```

Przeglądanie zawartości pliku

cat **opcje** **nazwa pliku** - wyświetla całą treść pliku na ekranie

Dostępne opcje:

- b** – numerowanie niepustych wierszy,
- n** – numerowanie wszystkich wierszy,
- s** – zastąpienie kilku kolejnych pustych wierszy jednym,
- v** – dodatkowe drukowanie znaków kontrolnych.
- Nie umożliwia stronicowania - gdy plik jest duży można obejrzeć tylko ostatni ekran tekstu.
- Ograniczeń tych nie mają polecenia **more** i **less**

more **opcje** **nazwa pliku**

less - bardzo podobne do **more**, wyświetla treść pliku i umożliwia przeglądanie go za pomocą strzałek ↑ ↓

Polecenia

head opcje pliki

powoduje wyświetlenie pierwszych 10 wierszy podanych plików - gdy jest wiele plików, nazwa pliku poprzedza jego zawartość

Dostępne opcje:

- cn[b|k|m]** – wyświetlenie pierwszych **n** bajtów, **b** bloków, **k** kilobajtów albo **m** megabajtów
- #** – zmiana liczby wierszy do wyświetlenia,
- q** – zakaz wypisywania nazwy pliku.

tail opcje pliki

powoduje wyświetlenie ostatnich 10 wierszy podanego pliku

Dostępne opcje:

- cn[b|k|m]** – wyświetlenie ostatnich **n** bajtów
- n** – wyświetlenie ostatnich **n** wierszy,
- v** – wyświetlenie nazwy pliku w wierszu tytułowym.

Polecenia

wc opcje pliki

zlicza liczby znaków, słów i wierszy w pliku tekstowym

Dostępne opcje:

- c – tylko liczba znaków,
- l – tylko liczba wierszy,
- w – tylko liczba słów.

wc -l ./przyklad

Polecenia

cut opcje pliki

umożliwia wycięcie serii pól lub kolumn z wiersza pliku wejściowego.

Dostępne opcje:

- b lista** – wybranie znaków z pozycji podanych na liście
- c lista** – wybranie kolumn podanych na liście
- f lista** – wybranie pól (oddzielonych tabulatorami albo znakami separatorów) z listy
- dznak separatora** – używane do podania znaku separatora pól

cut -d: -f1,3 /etc/passwd

wypisuje nazwy i identyfikatory wszystkich użytkowników

Sortowanie zawartości pliku

sort **opcje pliki** - sortowanie wierszy podanych plików

Dostępne opcje:

- +n-m—ustawienie klucza sortowania między polami n i m,
- b—ignorowanie spacji na początku wiersza,
- c—sprawdzenie, czy podane pliki nie są już posortowane. Jeżeli tak, to program kończy się komunikatem błędu,
- d—ignorowanie znaków interpunkcyjnych w czasie sortowania,
- f—ignorowanie małych/wielkich liter,
- i—ignorowanie niedrukowalnych znaków ASCII,
- m—połączenie dwóch plików wejściowych,
- n—sortowanie numeryczne,
- o plik—przeadresowanie standardowego urządzenia wyjściowego do pliku,
- r—odwrócenie kolejności sortowania,
- t znak—podanie znaku separatora kolumn (domyślnie spacja albo tabulacja),
- u—usunięcie powtarzających się wierszy.

Przeszukiwanie zawartości pliku

- Polecenie grep jest uniwersalnym programem przeznaczonym do wyszukiwania w pliku wierszy zawierających określony wzorzec
- Dane ze strumienia wejściowego lub z pliku wejściowego czytane są wierszami i wypisywane są te wiersze, które zawierają podany wzorzec.

grep opcje wzorzec plik

- Uniwersalność programu grep bierze się z możliwości tworzenia złożonych wzorców za pomocą znaków uogólniających. Znaczenie wybranych metaznaków:
 - . – jeden dowolny znak
 - [a-z] – litera z podanego zakresu
 - .* – zero lub więcej powtórzeń poprzedzającego elementu
 - + – jedno lub więcej powtórzeń poprzedzającego elementu

Przeszukiwanie zawartości pliku

Określenie położenie wzorca w wierszu:

`^wzorzec` – pasuje do linii rozpoczynających się danym wzorcem

`wzorzec$` – pasuje do linii kończących się danym wzorcem

Znak `^` po otwierającym nawiasie kwadratowym pełni rolę zaprzeczenia.

Np. `[^a-z]` oznacza znak nie będący małą literą.

Przykłady zastosowania wzorców polecenia `grep`:

`abc` – wypisuje wiersze zawierające łańcuch „abc”

`^abc` – wypisuje wiersze zaczynające się łańcuchem „abc”

`abc$` – wypisuje wiersze kończące się łańcuchem „abc”

`a..c` – wypisuje wiersze zawierające znaki „a” i „c”, rozdzielone dwoma dowolnymi znakami

Przenoszenie plików

Nazwę pliku można zmienić poleceniem:

mv opcje stara_nazwa nowa_nazwa

Opcja **-i** (zabezpieczenie przed nadpisaniem) sprawdza, czy plik o nowej nazwie istnieje - jeśli tak, pojawia się pytanie o zgodę lub nie na nadpisanie pliku:

mv -i stara_nazwa nowa_nazwa

Jeśli operacja **mv** wykonywana jest pomiędzy katalogami, plik zostaje przeniesiony do podanego katalogu

mv opcje nazwa_pliku nazwa_katalogu

mv opcje nazwa_pliku nazwa_katalogu/nowa_nazwa

Przenoszenie plików

- b**—utworzenie kopii zapasowej pliku przed zniszczeniem jego zawartości,
- f**—brak pytania o potwierdzenie przed zniszczeniem zawartości plików,
- u** —przeniesienie plików tylko wtedy, kiedy są nowsze niż pliki docelowe o tej samej nazwie

Można przenosić nie tylko pliki zwykłe, ale również całe katalogi.

mv katalog1 katalog2

katalog1 przeniesiony zostanie jako podkatalog katalogu **katalog2**

Kopiowanie plików

Polecenie **cp** tworzy kopię pliku źródłowego najpierw przez utworzenie pliku, a następnie skopiowanie do niego danych.

cp opcje plik_źródłowy plik_docelowy

Jeśli więc istnieje plik o tej nazwie, zostaje nadpisany:

cp -i plik_źródłowy plik_docelowy

- i** - powoduje, że polecenie **cp** najpierw sprawdzi, czy plik docelowy już istnieje- jeśli tak tak, użytkownik zostanie zapytany, czy nadpisać istniejący plik

Kopiowanie plików

Dostępne opcje

- b**—utworzenie kopii plików przed zniszczeniem ich zawartości,
- f**—wymuszenie kopiowania i zniszczenia zawartości istniejących plików,
- l** —utworzenie twardego dowiązania zamiast kopiowania,
- s**—utworzenie dowiązania symbolicznego zamiast kopiowania,
- v**—wyświetlenie nazwy każdego pliku w czasie kopiowania,
- r**—kopiowanie całych katalogów wraz z podkatalogami,
- p**—kopiowanie plików do katalogu docelowego z zachowaniem hierarchii podkatalogów.

Kopiowanie plików

- Można również kopiować i przenosić całe katalogi
- Polecenia **cp** oraz **mv** przyjmują nazwę katalogu jako pierwszy argument
- Aby skopiować katalog, polecenie **cp** wymaga opcji **-r**, co powoduje to rekursję, czyli kopiowanie katalogów wraz z podkatalogami i ich zawartością:

cp -r katalog1 katalog2

- **Katalog1** jest kopiowany jako podkatalog katalogu **katalog2**

Kopiowanie plików

- Można również kopiować i przenosić całe katalogi
- Polecenia **cp** oraz **mv** przyjmują nazwę katalogu jako pierwszy argument
- Aby skopiować katalog, polecenie **cp** wymaga opcji **-r**, co powoduje to rekursję, czyli kopiowanie katalogów wraz z podkatalogami i ich zawartością:

cp -r katalog1 katalog2

- **Katalog1** jest kopiowany jako podkatalog katalogu **katalog2**

Usuwanie plików

rm opcje plik1 plik2

Dostępne opcje:

- f** – usuwa pliki bez żadnego uprzedzenia,
 - r** – pozwala na usuwanie katalogu ze wszystkimi podkatalogami i plikami,
 - i** – żąda potwierdzenia dla każdego usuwanego pliku,
 - v** – wyświetla nazwy każdego kasowanego pliku
- Polecenie to jest nieodwracalne, nie można odtworzyć pliku po jego usunięciu

rm -i plik1 plik2

Usuwanie katalogów

- Pusty katalog, czyli katalog nie zawierający żadnych plików ani podkatalogów, można usunąć poleceniem:

rm katalog

- Katalog ze wszystkimi podkatalogami

rm -r katalog

- Najpierw usuwa całą zawartość katalogu, a następnie sam **katalog**

Przekierowania

- Wszystkie pliki Linuxa są logicznie zorganizowane jako ciągły strumień bajtów
- Fizycznie pliki są ułożone jako rozrzucone bloki w pamięci dyskowej
- Każdemu procesowi w chwili uruchomienia przypisane są automatycznie standardowe strumienie: wejściowy i wyjściowy
- Wejściowy ciąg danych jest w Linuxie nazwany standardowym wejściem (KLawIATURA), a wyjściowy (EKRAN) – standardowym wyjściem
- Standardowe wejście lub wyjście użytkownik może zmienić dzięki przekierowaniom, np. można skierować standardowe wyjście do pliku, a nie na ekran - należy umieścić operator przekierowania > i nazwę pliku w wierszu poleceń

Przekierowania

- Standardowe wyjście można przekierować do pliku np.

ls >alfa

- wynik polecenia **ls** zamiast na ekranie znajdzie się w pliku o nazwie **alfa** - jeśli plik o nazwie **alfa** (docelowy) istnieje, to zostaje nadpisany, jeśli nie istnieje, to zostanie utworzony.
- Do przekierowania można także używać operatora >>

ls >>alfa

- powoduje dopisanie do pliku **alfa** (gdy plik o takiej nazwie istnieje) wyniku polecenia **ls** - jeśli plik **alfa** nie istnieje, to zostanie utworzony.
- Operator przekierowania jest wykonywany wcześniej niż zasadnicze polecenie łączenia plików:

cat plik1 plik2 > plik3

plik1 zostaje połączony z plikiem **plik2** i zapamiętany w pliku **plik3**.

Przekierowania

- Do standardowych operacji strumieni można odwołać się przez ich numery:

0 – standardowe wejście (klawiatura),

1 – standardowe wyjście (monitor),

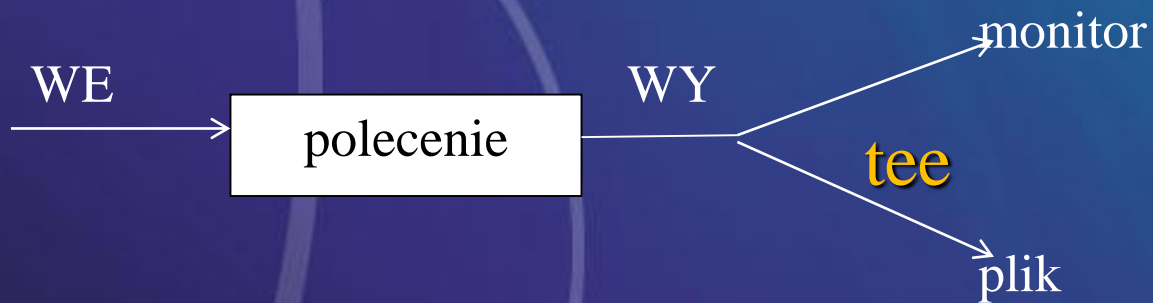
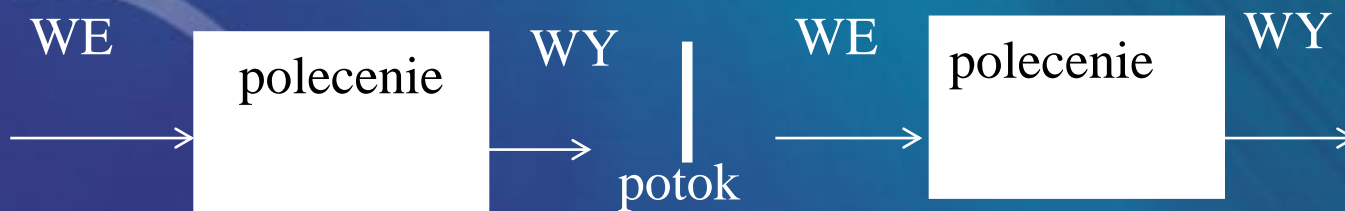
2 – standardowe wyjście błędów (monitor).

- Domyślne przekierowanie wyjścia błędu (2) działa na standardowym wyjściu (monitorze), czyli 1

cat test 2>blad

- spowoduje przekierowanie komunikatu o błędach na standardowy strumień określony cyfrą 2, czyli standardowe wyjście błędów - komunikat o błędzie pojawi się w pliku blad.

Potoki



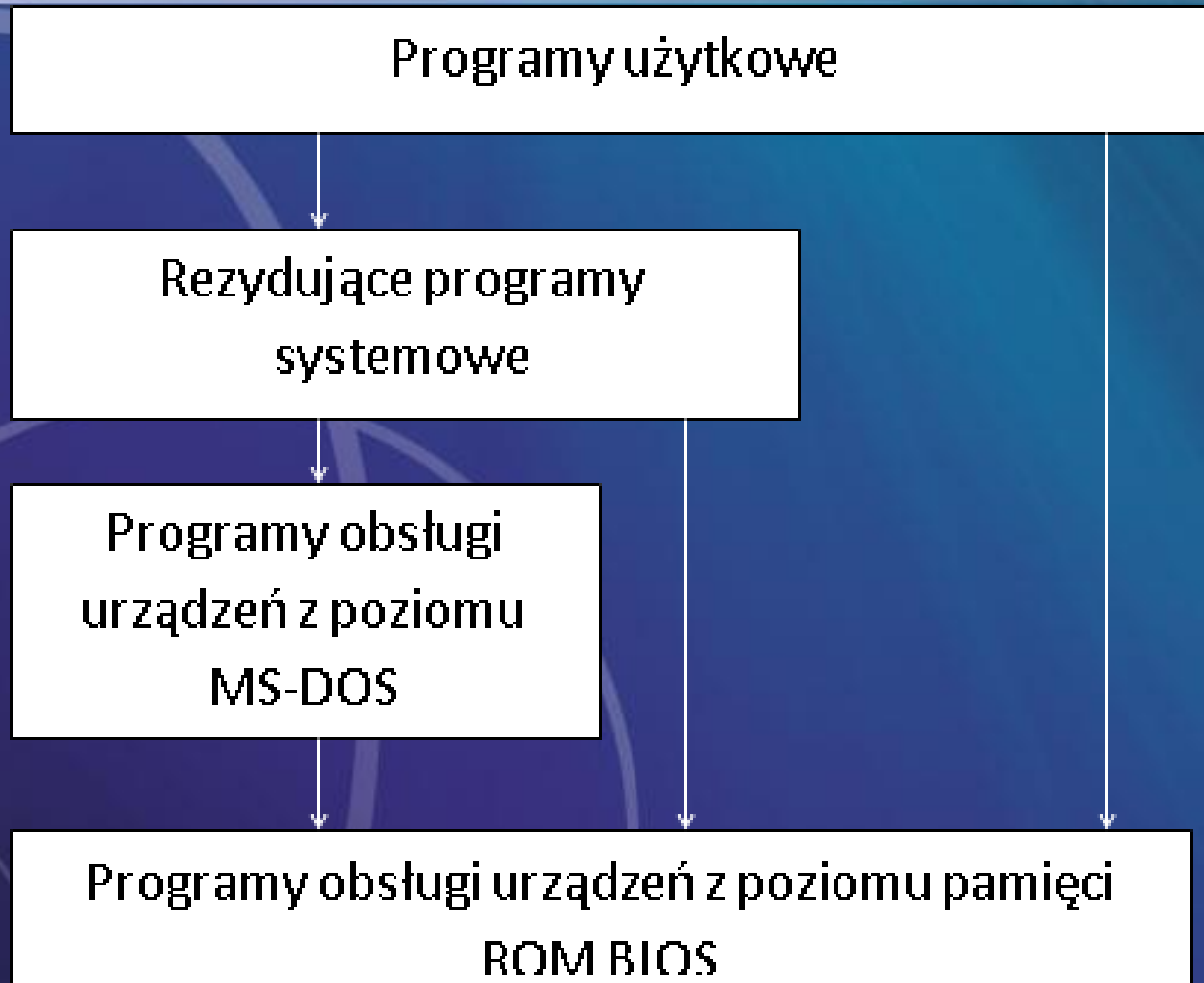
Struktury systemów operacyjnych

- System operacyjny, to bardzo złożony system informatyczny.
- Musi on mieć odpowiednią strukturę, tak aby jego zadania mogły być podzielone na mniejsze moduły o dobrze określonej funkcjonalności i interfejsach.

Trzy popularne struktury to:

- struktura jednolita,
- struktura warstwowa,
- struktura klient-serwer.

Struktura systemu MS DOS

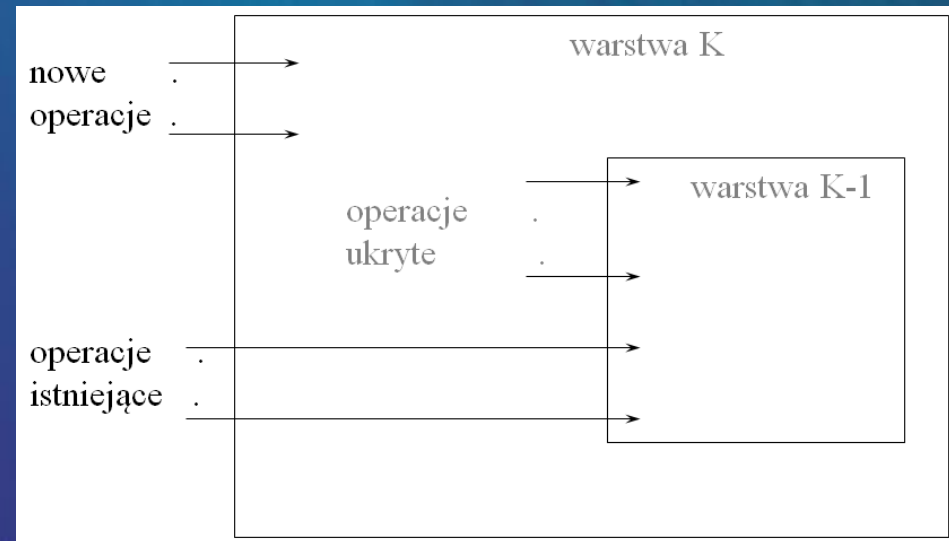


Struktura systemu UNIX

Użytkownicy		
programy shell i polecenia kompilatory i interpretery biblioteki systemowe		
Interfejs między funkcjami systemowymi a jądrem		
sygnały obsługa terminali system znakowego we/wy programy obsługi terminali	system plików wymiana system blokowego we/wy programy obsługi dysków i taśm	planowanie przydziału procesora zastępowanie stron stronicowanie na żądanie pamięć wirtualna
Interfejs między jądrem a sprzętem		
sterowniki terminali terminale	sterowniki urządzeń dyski i taśmy	sterowniki pamięci pamięć operacyjna

Warstwa systemu operacyjnego

- Tworzenie systemu polega na podzieleniu go na moduły, które są połączone w warstwy.
- Każda warstwa spełnia funkcje, które zależą tylko od warstwy znajdującej się pod spodem.
- Podział na moduły zmniejsza stopień wzajemnych zależności między różnymi składowymi systemu.
- Pozwala uniknąć niepożądanych powiązań.



Struktura klient - serwer

- Podział systemu operacyjnego na moduły.
- Moduły nie są rozmieszczone w warstwach .
- Nie komunikują się między sobą poprzez wywoływanie procedur, ale wysyłają komunikaty za pośrednictwem centralnego programu obsługi komunikatów.
- Komunikaty mogą być wysyłane w obie strony.
- Moduł wysyłający początkowy komunikat nazywa się **klientem**.
- Moduł odbierający ten komunikat **serwerem**.

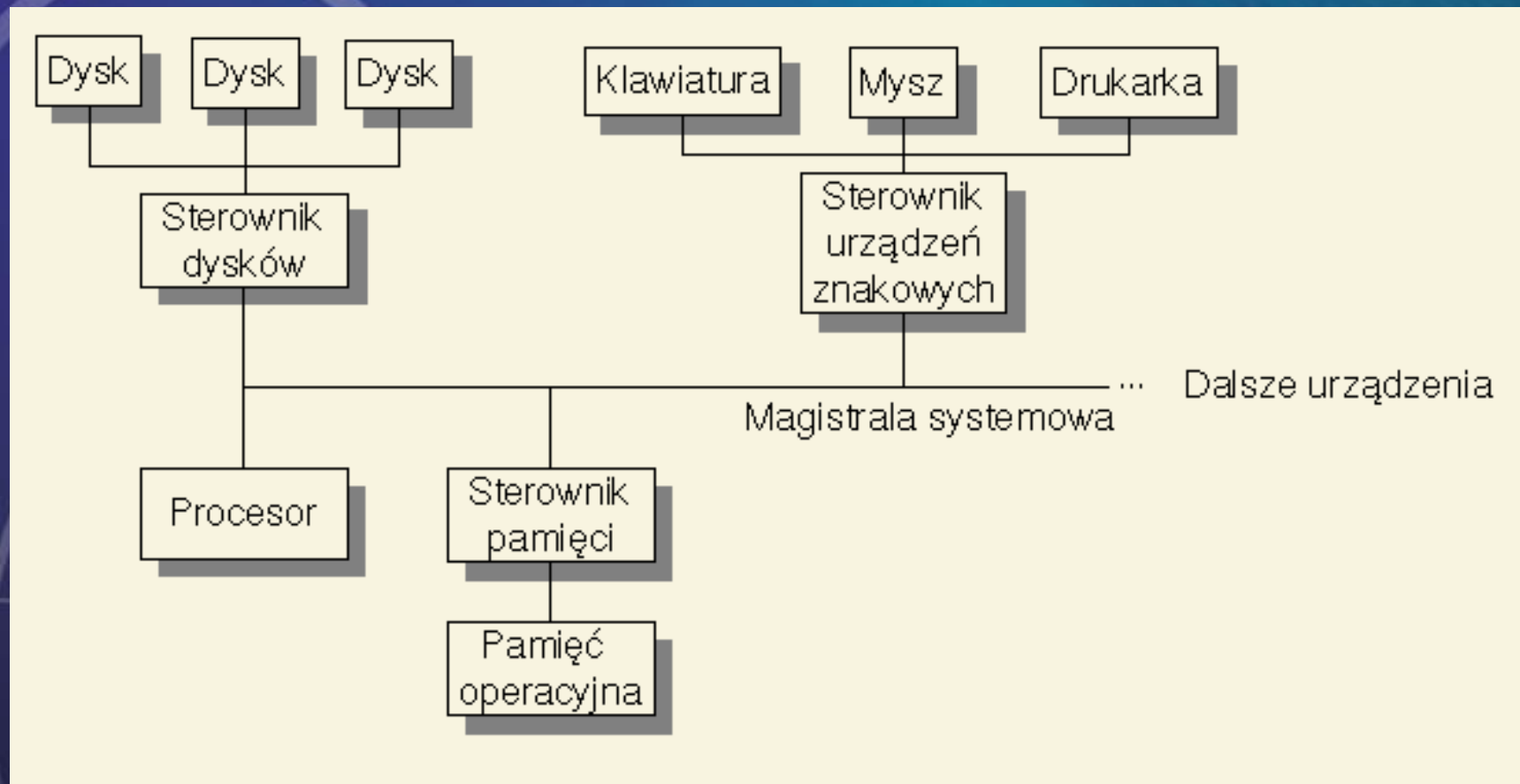
Przykładem systemu o takiej strukturze jest system Windows

Udogodnienia sprzętowe

- Aby programy użytkownika nie mogły dezorganizować pracy systemu, sprzęt musi mieć odpowiednie mechanizmy gwarantujące właściwe zachowanie się całości czyli tzw. **udogodnienia sprzętowe**:

- Mechanizm przerwań
- Ochrona pamięci operacyjnej
- Zbiór rozkazów uprzywilejowanych
- Zegar czasu rzeczywistego

System komputerowy z obsługą przewań



System komputerowy z obsługą przerwań

- Mechanizm przerwań służy informowaniu o zdarzeniach zachodzących w urządzeniach wejścia/wyjścia.
- Typowe takie zdarzenia to zakończenie zleconej operacji wejścia/wyjścia oraz zdarzenia zewnętrzne.
 - urządzenie generuje sygnał przerwania - w momencie wystąpienia zdarzenia,
 - procesor w odpowiedzi na ten sygnał przerywa wykonywanie ciągu instrukcji i wykonuje określoną procedurę obsługi przerwania,
 - po zakończeniu procedury obsługi przerwania następuje powrót do przerwanego ciągu instrukcji.

Mechanizm przerwań

W jaki sposób jedne przerwania są odróżniane od innych ?

- 1. odpytywanie** - procedura obsługi przerwania pyta się kolejno wszystkich urządzeń, czy to nie one spowodowały przerwanie.
- 2. wektor przerwań** - w systemie może być wiele rodzajów przerwań, a każdemu rodzajowi odpowiada osobna procedura obsługi - adresy tych procedur są przechowywane w tablicy systemowej nazywanej wektorem przerwań.

Synchroniczne i asynchroniczne WE/WY

synchroniczne wykonywanie operacji wejścia/wyjścia

- Procesor zleca wykonanie operacji i czeka, aż otrzyma potwierdzenie zakończenia operacji.

asynchroniczne wykonywanie operacji wejścia/wyjścia

- Procesor zleca wykonanie operacji, przerywa wykonywanie aktualnie wykonywanego zadania i nie czekając na zakończenie operacji wejścia/wyjścia zaczyna wykonywać inne zadanie.
- Po zakończeniu operacji wejścia/wyjścia urządzenie zewnętrzne generuje przerwanie.

Układ DMA

- Czas procesora zajmują przy asynchronicznych operacjach wejścia/wyjścia:
 - obsługa tablicy stanów urządzeń,
 - zlecenie wykonania operacji sterownikowi urządzenia,
 - przesłanie zapisywanych/odczytywanych danych,
 - obsługa przerwania kończącego wykonanie operacji.
- Jeśli przesyłane są duże ilości danych pomocny jest układ DMA (bezpośredniego dostępu do pamięci, ang. *direct memory access*).

Ochrona sprzętowa pamięci

- Ochrona całego SO przed wpływami programów użytkownika.
- Wzajemna ochrona programów użytkowników.
 - programy użytkowników ładowane są do przeznaczonych dla nich obszarów pamięci,
 - ochrona przestrzeni adresowej (rejestr bazowy i graniczny),
 - rejestr bazowy i graniczny może modyfikować tylko system operacyjny,
 - wygenerowanie adresu programu pracującego w trybie użytkownika, poza własną przestrzenią adresową kończy się pułapką obsługiwaną przez SO.