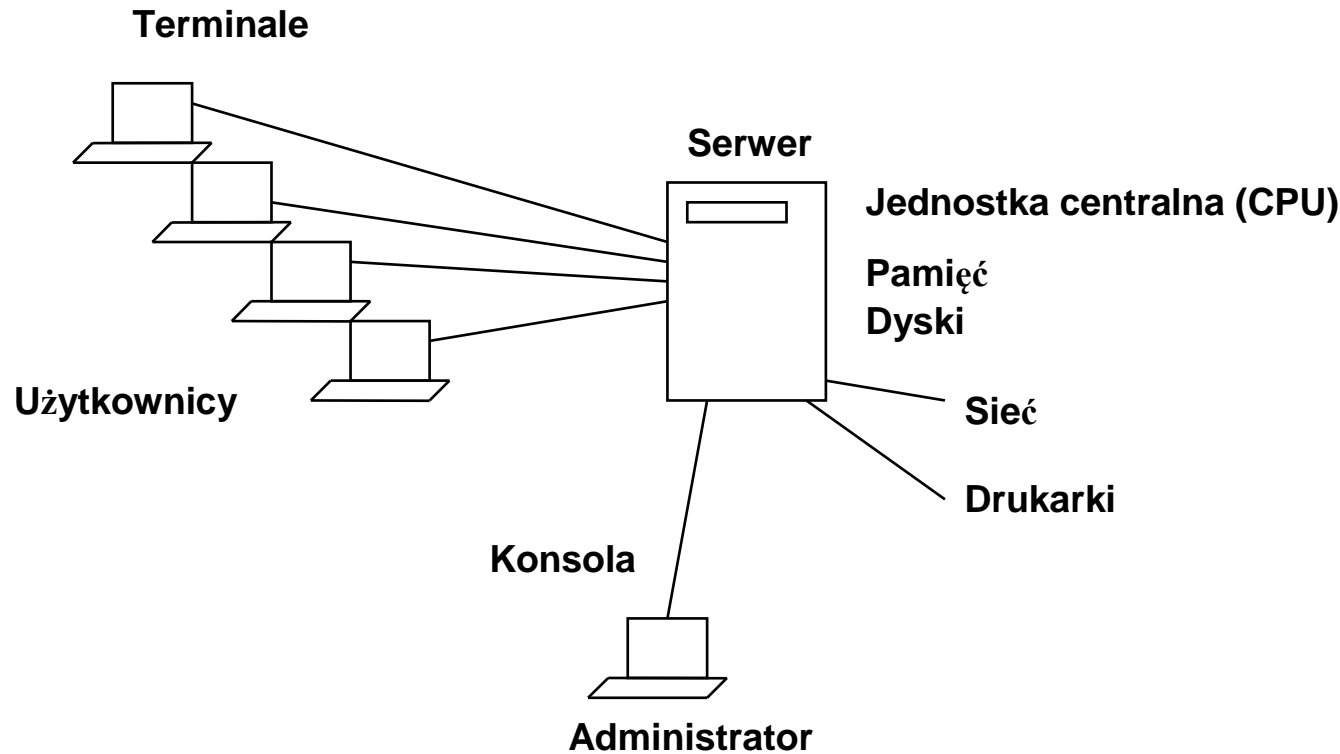


Przykładowa konfiguracja systemu komputerowego



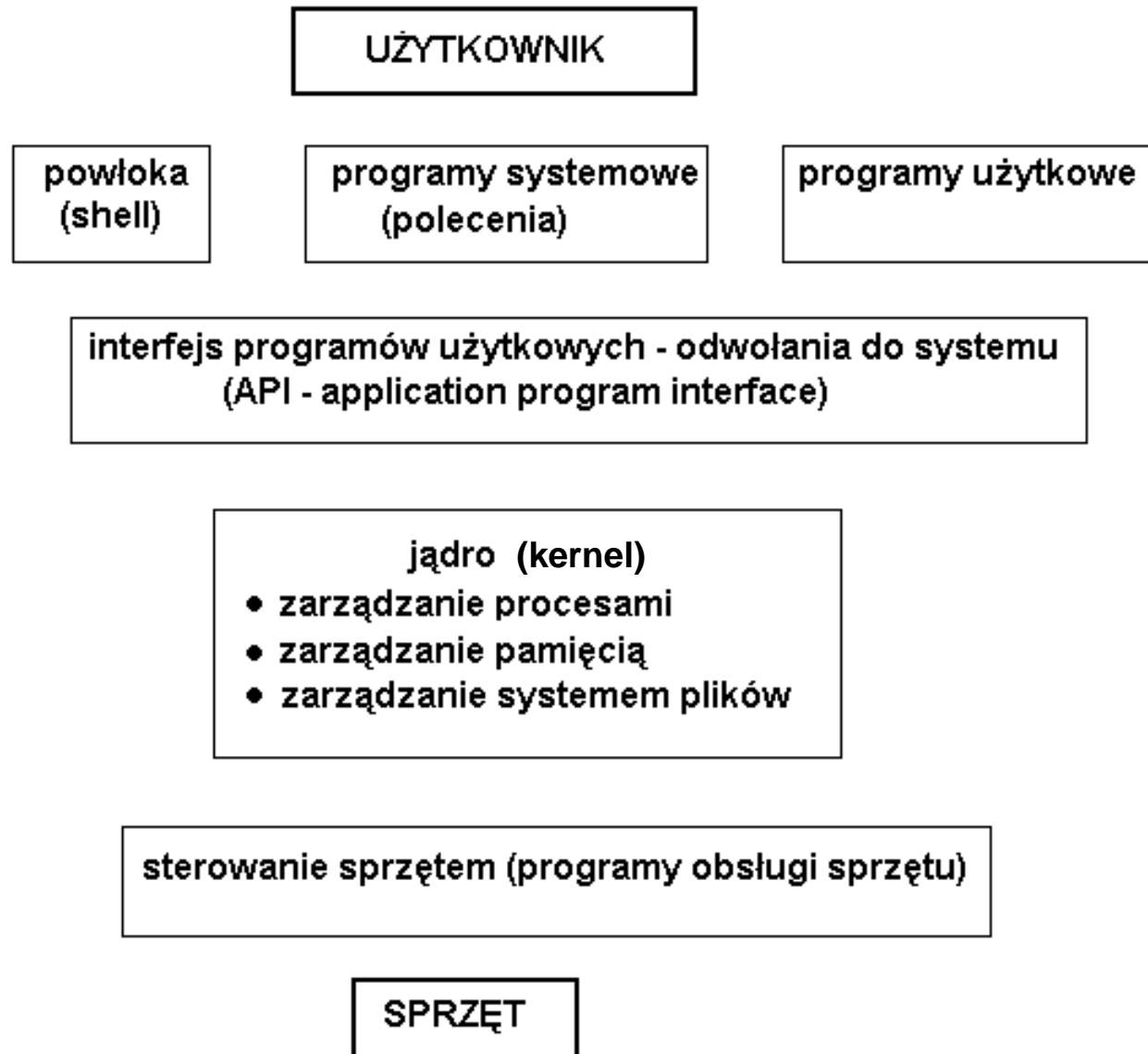
System operacyjny: program zarządzający pracą komputera

Zadania użytkownika wykonywane są jako procesy obsługiwane przez system operacyjny. System przydziela procesom zasoby komputera: dostęp do jednostki centralnej, pamięć, dyski, i inne.

Niezbędny administrator - osoba odpowiedzialna za sprawne działanie i bezpieczeństwo systemu. Również opiekun użytkowników, zakłada konta użytkowników, służy pomocą i radą.

Co to jest system operacyjny ?

Program pośredniczący między sprzętem komputerowym a użytkownikiem.



Krótką historia systemu operacyjnego UNIX

1969

AT&T Bell Labs

Ken Thompson, Ruddy Canaday, Joe Ossana opracowali pierwszy system UNIX

- Założenia: mały, ogólnego przeznaczenia, pracujący z podziałem czasu system operacyjny na minikomputerze PDP 7
- Cel budowy systemu: system operacyjny opracowany przez programistów dla programistów
- Podstawowe założenie: użytkownik wie co robi.

1973

4-ta edycja systemu AT&T, jądro systemu i Shell napisane w języku C opracowanym przez Denisa Ritchie.

1975

Zainteresowanie systemem UNIX na uniwersytetach

Dlaczego UNIX stał się popularny?

- mały, elastyczny, tani, dostępny na komputerach od micro do super komputerów
- właściwości zazwyczaj dostępne tylko w dużych systemach operacyjnych:
- architektura wieloprocusowa, inicjowanie asynchronicznych procesów
- hierarchiczny system plików

1975 Uniwersytet Kalifornijski

Początek prac nad systemem UNIX w Uniwersytecie Kalifornijskim w Berkeley
Berkeley Software Distribution (UNIX BSD)

lata 80-te Różne komercyjne wersje Unixa

1985 System V Interface Definition

1986 4.3 BSD

Duże firmy komputerowe zaczęły tworzyć własne systemy, zgodne z systemem UNIX. Systemy te łączyły w sobie cechy systemów AT&T oraz BSD.

Prace nad standardami - założenie Open System Foundation.

Nazwy systemu UNIX w różnych firmach

AT&T	UNIX	Apple	A/UX
Berkeley	BSD	DEC	Ultrix
HP	HP-UX	IBM	AIX
Microsoft	Xenix	SUN	SunOS, Solaris
Santa Cruz	UNIX SCO		

LINUX (LinusTorvalds)

open source linux - Powszechna Licencja Publiczna GNU

Można instalować oprogramowanie na dowolnej liczbie komputerów.

Dowolna liczba użytkowników może używać oprogramowania w tym samym czasie.

Można wykonać dowolną ilość kopii oprogramowania i przekazać je komukolwiek (redystrybucja "otwarta" lub "bez ograniczeń").

Brak ograniczeń w modyfikowaniu oprogramowania (z wyjątkiem zachowania w nietkniętym stanie pewnych uwag).

Nie ma ograniczeń w rozprowadzaniu, a nawet w sprzedaży programowania.

Dystrybucje (przykłady):

NOVEL SUSE, RED HAT, MANDRAKE, DEBIAN, SLACKWARE, ...

Wprowadzanie standardów

Cele standaryzacji

Budowa systemów tzw. "otwartych", charakteryzujących się:

- przenośnością aplikacji (portability),
- możliwością współpracy oprogramowania działającego na różnych maszynach (interoperability),
- skalowalnością (scalability), t.j. możliwością rozbudowy sprzętowej i rozbudowy aplikacji bez konieczności zmian systemu.

Istota standardu: określenie interfejsu, a nie implementacji.

Niektóre standardy i instytucje standaryzujące

SVID - standard firmy AT&T - pierwsza próba standardu interfejsu systemu operacyjnego.

POSIX - standard sponsorowany przez Instytut IEEE (Instytut of Electrical and Electronics Engineering). Prace nad tym standardem obejmują: interfejs systemu, programy shell, metody testowania zgodności danego systemu ze standardem, pracę w sieci, zagadnienia ochrony i inne.

OSF (Open System Foundation) stowarzyszenie wiodących firm: HP, IBM, DEC i innych, zajmujące się promocją systemów otwartych. Opracowany został standard OSF/Motif.

Aktualne prace obejmują między innymi: Distributed Computing Environment - przetwarzanie w środowisku rozproszonym, Distributed Management Environment - centralne zarządzanie sieciami heterogenicznymi.

ISO - International Standard Organization - koordynuje tworzenie i stosowanie standardów w skali międzynarodowej. Wprowadziła siedmiowarstwowy model odniesienia OSI (Open System Interconnection) dotyczący pracy w sieciach komputerowych.

Cechy systemu UNIX

System wielodostępny (ang. <i>multiuser</i>)	wielu użytkowników pracujących z jednym komputerem, wrażenie samodzielnej pracy
Praca interakcyjna (<i>interactive</i>)	użytkownik wprowadza polecenie z terminala, UNIX wykonuje polecenie, wyświetla wyniki i czeka na nowe polecenie
System wielozadaniowy (<i>multitasking</i>)	każdy użytkownik może uruchomić jednocześnie wiele programów
Zapewnia bezpieczeństwo (<i>security, privacy</i>)	identyfikacja użytkowników, ochrona dostępu do plików i katalogów, ochrona procesów
Niezależny od urządzeń system we/wy (<i>device independent I/O</i>)	każde urządzenie reprezentowane jest przez plik specjalny systemu
Komunikacja między procesami (ang. <i>interprocess communication</i>)	aplikacje mogą wzajemnie się ze sobą komunikować
System sieciowy (<i>networking</i>)	wbudowane jest oprogramowanie umożliwiające pracę w sieci
Polecenie/Program użytkowy (<i>commands/utilities</i>)	polecenie jest programem użytkowym, można samemu je napisać
Interpretator poleceń (<i>shell</i>)	jest wiele programów shell'a, można go zmienić

Struktura systemu operacyjnego

Podsystemy wykonujące zadania:

Zarządzanie procesami: tworzenie, usuwanie, zawieszanie, odwieszanie procesów, mechanizmy synchronizacji procesów, komunikacja między procesami.

Zarządzanie pamięcią: zarządzanie pamięcią główną, obszarem wymiany (swap), pojęcie pamięci wirtualnej.

Zarządzanie przestrzenią dyskową: zarządzanie wolną przestrzenią dysków, procesami, zapisywania informacji na dysku, szeregowanie zadań zapisu i odczytu.

Zarządzanie operacjami we/wy: obejmuje podsystem buforowania, interfejs: urządzenia -sterowniki, sterowniki urządzeń.

Zarządzanie plikami: tworzenie, usuwanie plików i katalogów, elementarne operacje z plikami i katalogami.

Podsystem ochrony: ochrona procesów przed działaniem innych procesów, mechanizmy zapewniające, że pliki, segmenty pamięci, CPU, inne zasoby są udostępnione tylko tym procesom, które mają autoryzację systemu operacyjnego. Ogólnie: mechanizmy kontroli dostępu programów, procesów, użytkowników do zasobów systemu komputerowego.

Praca sieciowa: usługi umożliwiające komunikację w sieci, pojęcie systemów rozproszonych.

Usługi systemu operacyjnego

Wykonywanie programów

Operacje we/wy

Operacje obsługi systemu plików

Komunikacja między procesami

Detekcja błędów

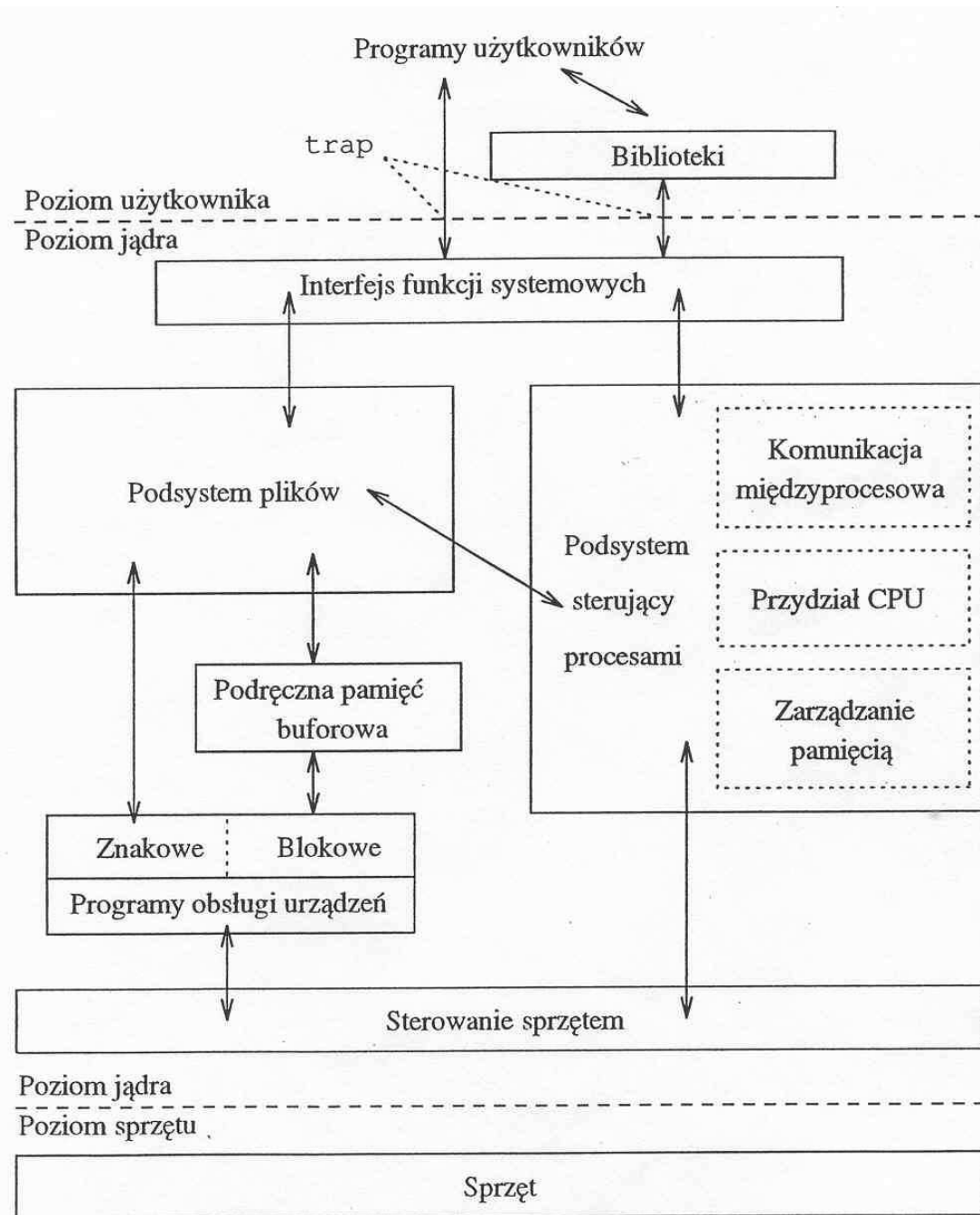
Przydział zasobów

Rozliczanie użytkowników (accounting)

Ochrona

Funkcje systemowe (system calls):

interfejs między procesami i systemem operacyjnym



Systemy plików

Pliki to jednostki logiczne przechowywanej informacji, niezależne od właściwości fizycznych urządzeń pamięciowych. Zwykle w plikach przechowywane są programy lub dane (tekst, liczby, grafika, itp.).

System plików - zbiór typów danych, struktur danych oraz funkcji systemowych (ang. *system calls*) używanych przez system operacyjny w celu przechowywania informacji w urządzeniach pamięci masowej (głównie na dyskach).

W systemach wielodostępnych systemy plików mają strukturę katalogową (hierarchiczną).

Pliki identyfikuje się za pomocą nazw. W systemie UNIX, w zależności od implementacji, pliki mogą mieć krótkie nazwy, do 14 znaków, lub długie nazwy, do 255 znaków.

Zestaw znaków dopuszczalnych obejmuje:

małe lub duże litery, cyfry, znaki specjalne takie jak „+, - , _ ' .”

Przykład

Dopuszczalne nazwy plików: .profile .xyz.abcd abc AbC 123..456..78 -a

UWAGA: Tej ostatniej nazwy lepiej jednak nie używać.

UWAGA: W systemie UNIX pliki mogą być identyfikowane za pomocą wielu różnych nazw (dowiązań).

Podstawowe typy plików:

- pliki zwykłe,
- pliki specjalne,
- katalogi,
- dowiązania symboliczne,
- potoki nazwane FIFO (ang. named pipe),
- gniazda (ang. UNIX--domain sockets).

W plikach zwykłych przechowujemy programy, dane, teksty, grafikę, itp. W systemie UNIX pliki zwykłe nie mają ustalonego formatu. Plik zwykły jest po prostu ciągiem bajtów o danej długości. Oczywiście aplikacje mogą tworzyć pliki o ściśle ustalonym formacie, na przykład plik assembler generuje plik, w którym wyróżniamy: nagłówek, kod wykonywalny, inicjalizowane dane.

Pliki specjalne, nazywane również plikami urządzeń zapewniają łączność z urządzeniami, na przykład z dyskami, terminalami, napędami taśmy, itp. W plikach specjalnych nie przechowuje się żadnych danych. Pliki te charakteryzują sposób działania urządzenia, wskazują miejsce podłączenia urządzeń do systemu oraz zapewniają dostęp do programów obsługi urządzeń („drajwerów”).

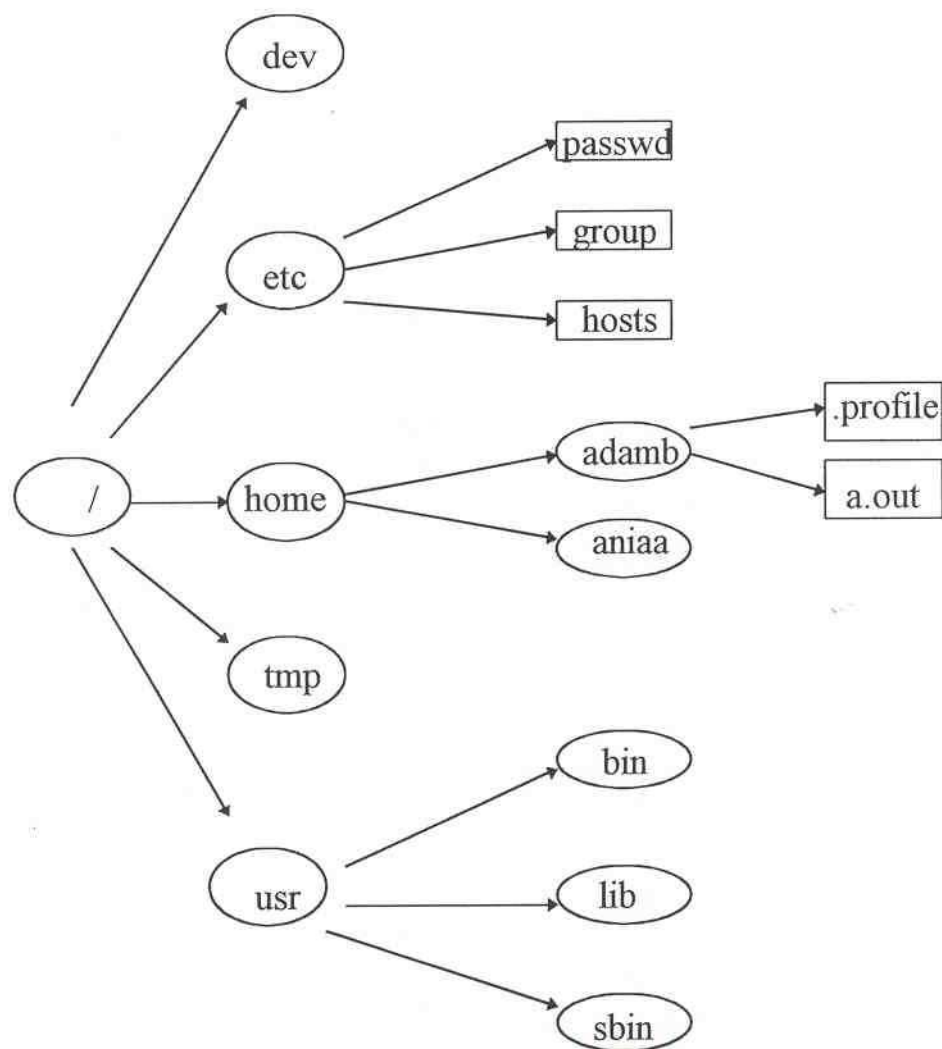
Katalogi służą do powiązania nazw plików z danymi znajdującymi się na dysku. W każdym katalogu może znajdować się pewna liczba plików i innych katalogów (podkatalogów). Katalog jest przechowywany jak plik zwykły i (w uproszczeniu) ma postać tabeli o dwóch kolumnach. Każdy wiersz tej tabeli zawiera nazwę pliku znajdującego się w katalogu (lub podkatalogu) oraz pewien numer, pozwalający na odszukanie atrybutów pliku i danych, które się w nim znajdują.

Dowiązania zwykłe mogą być tworzone w obrębie tego samego systemu plików. **Dowiązań symbolicznych** używa się ponad granicami systemów plików oraz w odniesieniu do katalogów. Przechowywane są w nich ścieżki dostępu do plików lub katalogów, na które dowiązania te wskazują.

Potoki nazwane (FIFO) wykorzystywane są do komunikacji między procesami. Do tworzenia tych potoków wykorzystywane są odpowiednie procedury biblioteczne. Procesy mogą otwierać potoki nazwane do odczytu i zapisu, tak jak otwierają pliki zwykłe.

Gniazda wprowadzone zostały w systemie BSD UNIX. Są wykorzystywane również do komunikacji między procesami. Wykorzystują jednak inne mechanizmy niż potoki nazwane.

Struktura (drzewo) katalogów typowa dla systemu UNIX



Ścieżka dostępu

Jeden z katalogów zawsze służy użytkownikowi pracującemu w systemie UNIX jako katalog bieżący.

Położenie pliku lub katalogu w drzewie katalogów określa ścieżka dostępu do pliku lub katalogu.

Bezwzględna ścieżka dostępu określa położenie pliku lub katalogu względem katalogu głównego (ang. root directory) /, na przykład:

```
/etc/passwd  
/home/adamb/.profile
```

Względna ścieżka dostępu określa położenie pliku lub katalogu względem katalogu bieżącego, na przykład (jeśli katalogiem bieżącym jest /home):

```
adamb  
adamb/a.out  
../usr/lib
```

Operacje dotyczące katalogów

zmiana katalogu bieżącego, `cd [ścieżka dostępu do katalogu]`

ustalenie nazwy katalogu bieżącego, `pwd`

sprawdzenie zawartości katalogu, `ls [-opcje] [ścieżka dostępu do katalogu]`

tworzenie katalogów, `mkdir [-opcje] [ścieżka dostępu do katalogu]`

usuwanie katalogów `rmdir [-opcje] [ścieżka dostępu do katalogu]`

utworzenie dowiązania do katalogu: `ln -s stara_nazwa nowa_nazwa`

Operacje odnoszące się do plików:

wypisanie zawartości pliku tekstowego: `cat [ścieżka dostępu do pliku]`
`more [ścieżka dostępu do pliku]`

drukowanie pliku: `lp [-opcje] [ścieżka dostępu do pliku]`

wypisanie atrybutów pliku: `ls [-opcje] [ścieżka dostępu do pliku]`

kopiowanie pliku: `cp [-opcje] co_kopiujemy dokąd_kopiujemy`

usuwanie pliku: `rm [-opcje] nazwa_pliku_lub_katalogu`

zmiana nazwy pliku lub jego przeniesienie:

`mv [-opcje] stara_nazwa nowa_nazwa`

utworzenie dowiązania do pliku lub katalogu:

`ln [-opcje] stara_nazwa nowa_nazwa`

Atrybuty plików i katalogów:

typ pliku,

prawa dostępu do pliku,

liczba dowiązań do pliku,

identyfikator właściciela,

identyfikator grupy,

rozmiar pliku w bajtach,

czas ostatniej modyfikacji pliku,

czas ostatniego dostępu do pliku,

czas ostatniej zmiany informacji w i-węźle,

nazwa pliku.

Przykład:

`$ ls -ld /etc`

`drwxr-xr-x 22 root root 1024 Sep 1995 /etc`

Każdemu plikowi przyporządkowany jest i-węzeł (ang. *i.node*), który jest rekordem przechowującym większość informacji o pliku.

Zawartość i-węzła:

- typ pliku,
- prawa dostępu do pliku,
- liczba dowiązań do pliku,
- identyfikator właściciela,
- identyfikator grupy,
- rozmiar pliku w bajtach,
- czas ostatniej modyfikacji pliku,
- czas ostatniego dostępu do pliku,
- czas ostatniej zmiany informacji w i.węźle,
- 12 wskaźników zawierających adresy bloków z danymi pliku (bloki bezpośrednio adresowane),
- wskaźnik zawierający adres bloku, w którym przechowywane są adresy bloków z danymi (adresowanie pośrednie jednostopniowe),
- wskaźnik zawierający adresy bloków, w których przechowywane są adresy bloków z adresami bloków z danymi (adresowanie pośrednie dwustopniowe),
- wskaźnik wykorzystywany w adresowaniu pośrednim trzystopniowym.

i-węzły są tworzone wtedy, gdy tworzony jest system plików.

Liczba i-węzłów w systemie plików zależy od jego rozmiaru oraz założonego średniego rozmiaru pliku (np. 2kB lub 6kB).

Każdy i-węzeł zajmuje 128 bajtów. i-węzły tworzą tablicę i-węzłów.

Poszczególne i-węzły identyfikowane są przez numery, określające ich położenie w tablicy i-węzłów.

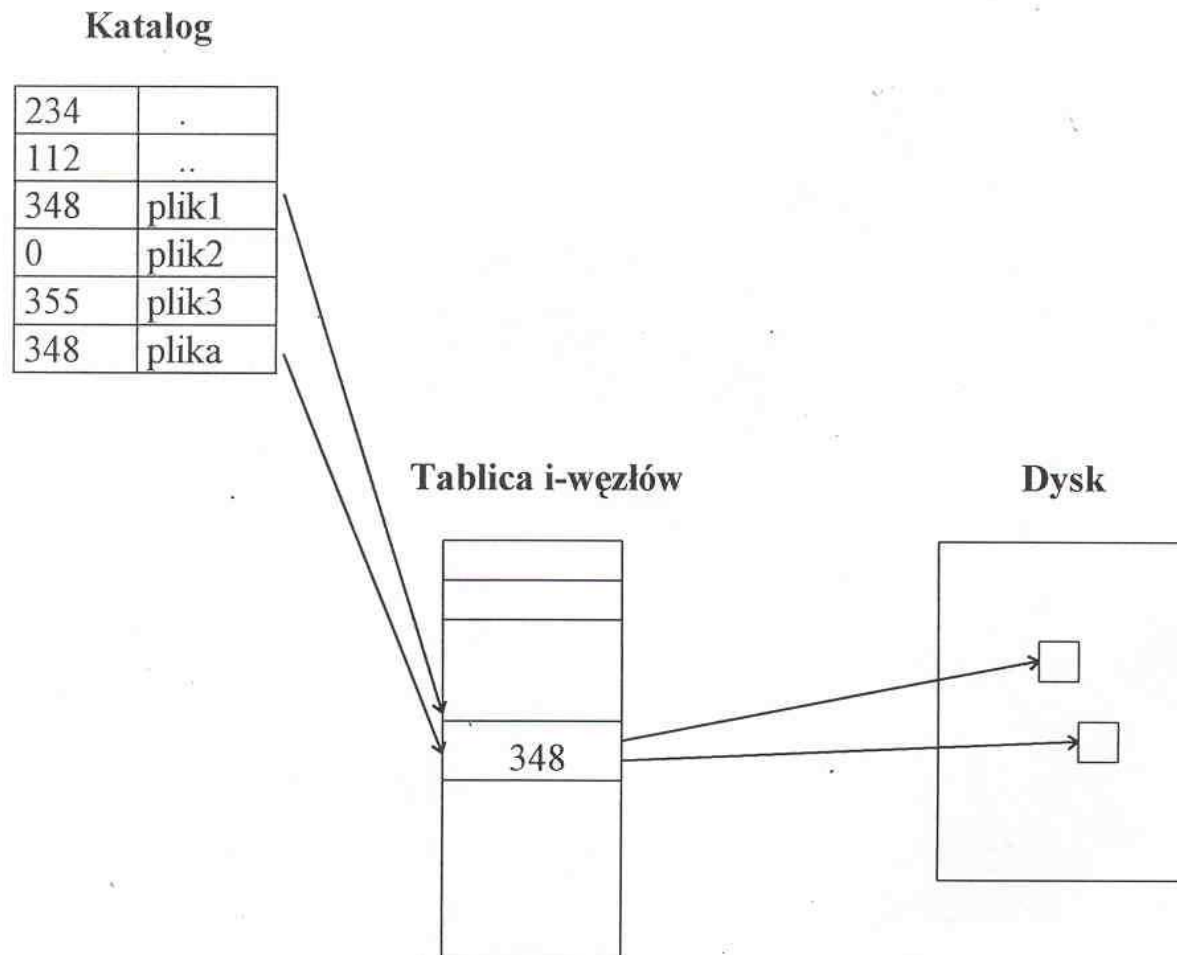
Aby sprawdzić, jaki i-węzeł został przyporządkowany danemu plikowi, należy w poleceniu `ls` użyć opcji `-li`.

Przykład:

```
$ ls -li /etc
```

```
77 drwxr-xr-x 22 root root 1024 Sep 1995 /etc
```

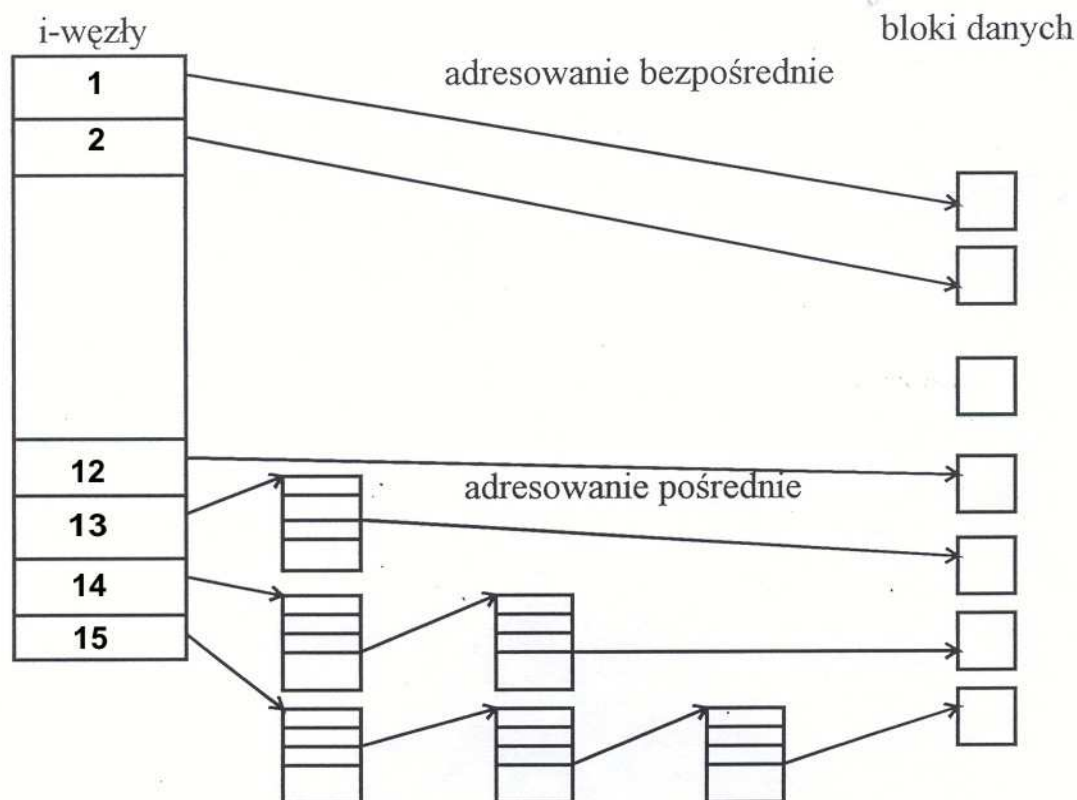
Nazwy plików są przechowywane w katalogach, łącznie z numerami odpowiadających tym plikom i-węzłów. Dzięki temu możliwe jest odczytanie atrybutów pliku oraz odszukanie przechowywanych w nim danych.



Blok identyfikacyjny (ang. Superblock) zawiera między innymi:

1. rozmiar systemu plików,
2. liczbę wolnych bloków w systemie plików,
3. listę wolnych bloków dostępnych w systemie plików,
4. indeks następnego wolnego bloku na liście wolnych bloków,
5. rozmiar tablicy i-węzłów,
6. liczbę wolnych i-węzłów w systemie plików,
7. listę wolnych i-węzłów w systemie plików,
8. indeks następnego wolnego i-węzła na liście wolnych i-węzłów.

Adresowanie bloków danych



Adresowanie tylko bezpośrednio nie jest efektywne i znacznie ograniczyłoby rozmiary plików.

W systemie UNIX zastosowano następujące reguły:

- tablica adresów przechowywana w i-węźle ma 15 elementów (wskaźników) i każdy zajmuje 4 bajty,
- 12 pierwszych wskaźników zawiera adresy bloków z danymi,
- następny, 13 wskaźnik zawiera adres bloku, w którym znajdują się adresy bloków z danymi,
- 14 wskaźnik to adres bloku, w którym umieszczane są adresy bloków zawierających adresy bloków z danymi,
- 15 wskaźnik to adres bloku, w którym umieszczane są adresy bloków przeznaczonych na adresy następnych bloków zawierających adresy bloków z danymi.

Przykład: jeśli blok zajmuje 4 kB, to

- adresowanie bezpośrednio pozwala na zaadresowanie danych plików o rozmiarach nie przekraczających: 48 kB,
- adresowanie pośrednio pozwala na zaadresowanie danych plików o rozmiarach nie przekraczających: $48 \text{ kB} + 1024 \cdot 4 \text{ kB} = 4144 \text{ kB}$,
- podwójne adresowanie pośrednio pozwala na zaadresowanie danych plików o rozmiarach nie przekraczających: $48 \text{ kB} + 1024 \cdot 4 \text{ kB} + 1024 \cdot 1024 \cdot 4 \text{ kB} = 4198448 \text{ kB}$,
- adresowanie pośrednio pozwala na zaadresowanie danych plików o rozmiarach nie przekraczających: $48 \text{ kB} + 1024 \cdot 4 \text{ kB} + 1024 \cdot 1024 \cdot 4 \text{ kB} + 1024 \cdot 1024 \cdot 1024 \cdot 4 \text{ kB}$,