



Systemy operacyjne

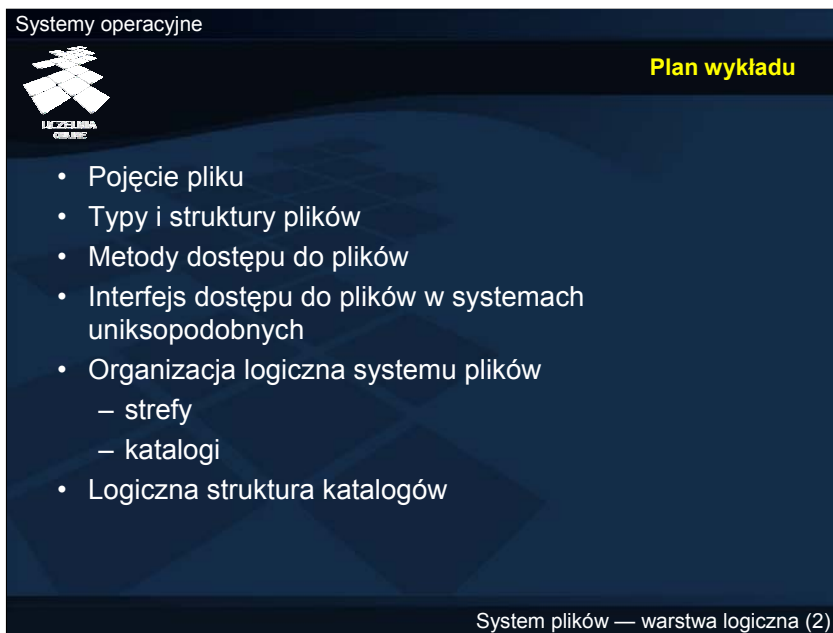
System plików — warstwa logiczna

Wykład prowadzą:
Jerzy Brzeziński
Dariusz Wawrzyniak

UCZELNIA
ONLINE

Nazwa modułu

Celem wykładu jest pokazanie systemu plików, jako abstrakcyjnego obrazu informacji, przechowywanej i udostępnianej przez system operacyjny. W zakresie tym mieszczą się podstawowe pojęcia związane z plikami (pojęcie samego pliku, typu oraz struktury) oraz ich logiczną organizacją (strefy i katalogi). Kluczowym elementem tego obrazu jest sposób dostępu do zawartości pliku.



Systemy operacyjne

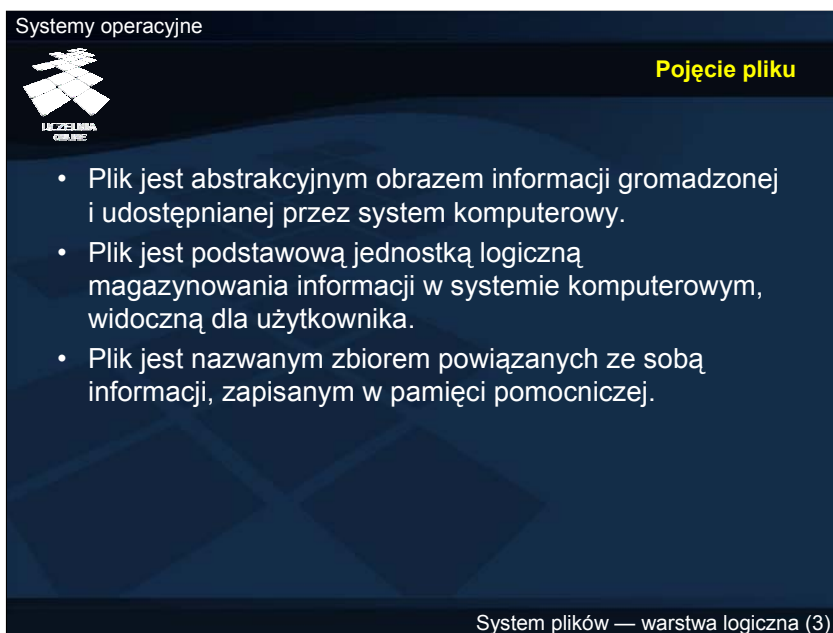
UCZELNIA
ONLINE

Plan wykładu

- Pojęcie pliku
- Typy i struktury plików
- Metody dostępu do plików
- Interfejs dostępu do plików w systemach uniksopodobnych
- Organizacja logiczna systemu plików
 - strefy
 - katalogi
- Logiczna struktura katalogów

System plików — warstwa logiczna (2)

Wykład rozpoczyna wprowadzenie do systemu plików, obejmującym zdefiniowanie podstawowych pojęć (plik, typ, struktura), atrybutów pliku oraz roli i zadań systemu operacyjnego w odniesieniu do systemu plików. Następnie omawiane są metody dostępu do plików oraz przykładowy interfejs operacji plikowych, ze szczególnym zwróceniem uwagi na sposób realizacji metod dostępu. Na końcu omawiana jest organizacja logiczna systemu plików, czyli ten element, który kształtuje obraz organizacji informacji, postrzegany przez użytkownika.



Systemy operacyjne

Pojęcie pliku

- Plik jest abstrakcyjnym obrazem informacji gromadzonej i udostępnianej przez system komputerowy.
- Plik jest podstawową jednostką logiczną magazynowania informacji w systemie komputerowym, widoczną dla użytkownika.
- Plik jest nazwanym zbiorem powiązanych ze sobą informacji, zapisanym w pamięci pomocniczej.

System plików — warstwa logiczna (3)

Pojęcie *pliku* (ang. file) jest bardzo ogólne i może być różnie rozumiane w zależności od punktu widzenia. Pojęcie to, podobnie jak *system operacyjny*, czy *proces*, trudno zdefiniować precyzyjnie w krótkiej i zwartej formie.

Intuicyjnie plik jest ciągiem danych (bitów, bajtów, rekordów itp.), których znaczenie (semantykę) określa jego twórca i jego użytkownik. Np. użytkownik, tworząc plik z programem w języku C, określa, że jest to plik, na podstawie którego kompilator potrafi wygenerować kod pośredni, a po dołączeniu odpowiednich bibliotek konsolidator (linker) potrafi wygenerować plik z programem binarnym.

Zamieszczone definicje podkreślają aspekt zewnętrzny (obraz informacji, jednostka magazynowania) oraz wewnętrzny (powiązane informacje).

Systemy operacyjne



Zadania systemu operacyjnego

- Zadaniem systemu operacyjnego w odniesieniu do plików jest zapewnienie odwzorowania pomiędzy abstrakcyjnym obrazem informacji a jego reprezentacją na urządzeniu fizycznym.
- Wyszczególnienie zadań:
 - identyfikacja pliku (hierarchiczna struktura katalogów),
 - udostępnienie interfejsu operacji plikowych (API),
 - realizacja operacji dostępu do plików i katalogów z zapewnieniem bezpieczeństwa (synchronizacja i autoryzacja dostępu), spójności i efektywności.

System plików — warstwa logiczna (4)

System operacyjny udostępnia pliki w jakiejś postaci logicznej (zwanej też wirtualną), np. sekwencji rekordów, tablicy bajtów itp., tzn. udostępnia operacje, z pomocą których można odpowiednio manipulować takim właśnie jednostkami. Wykonanie takiej operacji musi zostać przełożone na operacje dostępu do zawartości pliku, zgodnie z jego fizyczną strukturą, np. do zawartości sektora na dysku.

Zlecenie operacji na pliku wymaga jego zidentyfikowania. Dostarczanie wygodnych dla użytkownika identyfikatorów, to jedno z zadań systemu operacyjnego. Drugim zadaniem jest dostarczenie funkcjonalnie kompletnego interfejsu operacji plikowych oraz ich implementacji w celu realizacji dostępu.

Systemy operacyjne



Atrybuty pliku

- Nazwa — ciąg znaków służących użytkownikowi do identyfikacji pliku
- Typ — informacja służąca do rozpoznania rodzaju zawartości pliku i tym samym sposobu interpretacji
- Lokalizacja — informacja służąca do odnalezienia pliku w systemie komputerowym (urządzenie i położenie pliku w tym urządzeniu)
- Rozmiar — bieżący rozmiar pliku w ustalonych jednostkach (bajtach, słowach, blokach itp.)
- Ochrona — informacje umożliwiające kontrolę dostępu
- Czasy dostępu — daty i czasy wykonywania pewnych operacji na pliku, typu odczyt, modyfikacja, utworzenie

System plików — warstwa logiczna (5)

Nazwa pliku tworzona jest dla wygody użytkownika. System operacyjny ma najczęściej jakiś inny, wewnętrzny mechanizm identyfikacji pliku (np. numer i-węzła w systemie UNIX, referencja w NTFS).

Typ pliku, jako atrybut, istotny jest w systemach, które rozróżniają typy plików na poziomie jądra systemu operacyjnego. W praktyce pewne typy plików muszą być rozpoznawane przez jądro, np. pliki z programem dla procesu.

Lokalizacja jest atrybutem które w istotnym stopniu decyduje o różnicach w implementacji systemu plików.

Kontrola dostępu polega na weryfikacji uprawnień do wykonania operacji, żądanej przez użytkownika, np. odczytu pliku, zapisu pliku itp.

Czasy dostępu umożliwiają działanie niektórych narzędzi programistycznych, np. programu `make`. Przydatne są też czasami przy wyszukiwaniu pliku.

Systemy operacyjne



Typy plików

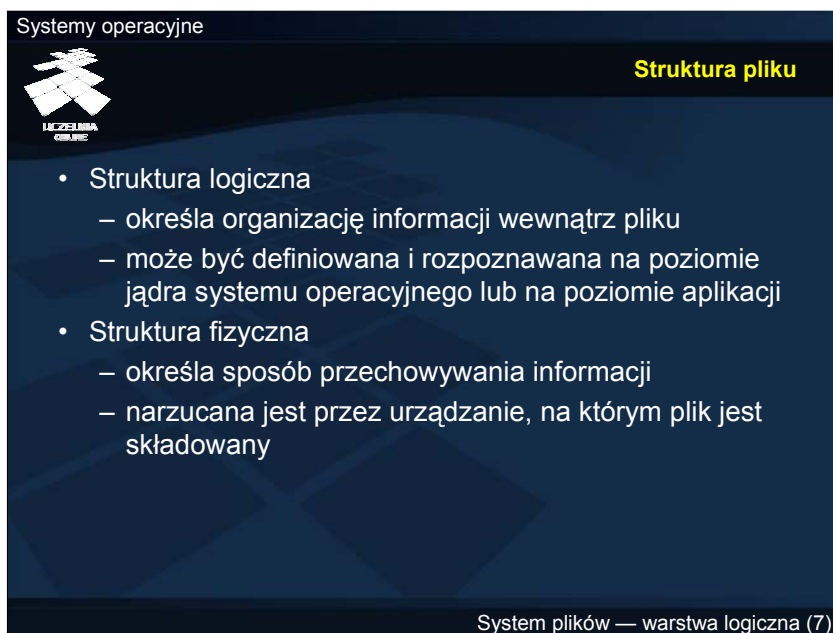
- Typ pliku określa rodzaj informacji przechowywanej w pliku i tym samym sposób interpretacji jego zawartości, np. program binarny, wynik kompilacji, kod źródłowy, makrodefinicja (plik wsadowy, skrypt powłoki itp.), tekst, biblioteka programisty, grafika, dane aplikacji.
- Informacja o typie pliku może być przechowywana w strukturach wewnętrznych systemu plików, w zawartości samego pliku, w katalogach lub w nazwie pliku.
- Typ pliku może być rozpoznawany przez system operacyjny, ale może to być również tylko informacja interpretowana przez użytkownika lub aplikację.

System plików — warstwa logiczna (6)

Problem rozpoznawania typów pliku można pozostawić użytkownikowi, wspomagając go ewentualnie przez przyjęcie pewnej konwencji nazewnicznej.

Jeśli typ pliku rozpoznawany jest przez jądro systemu operacyjnego, to możliwa jest optymalizacja dostępu, zabezpieczenie przed popełnieniem pewnych błędów, związanych z niewłaściwą interpretacją zawartości pliku (np. próba wyświetlenia zawartości pliku z programem binarnym) itp.

W systemie UNIX, w którym większość zasobów reprezentowana jest przez pliki, wyróżnia się następujące typy: plik zwykły, katalog, dowiązanie symboliczne, urządzenie blokowe, urządzenie znakowe, łącze nazwane, gniazdo. Rozróżnienie to jest bardziej związane z rolą pliku w systemie niż z jego zawartością. Jeśli chodzi o plik zwykły, to nie ma on typu — z punktu widzenia jądra systemu operacyjnego jest to ciąg bajtów, niezależnie od rodzaju zawartości. Jedynie pliki zwykłe z prawem *eXecute* są w pewien sposób wyróżnione, tzn. muszą mieć odpowiednią strukturę (np. COFF lub ELF), wymaganą przez jądro systemu operacyjnego w celu prawidłowego uruchomienia procesu (chyba że są to skrypty powłoki). Jądro dostarcza zestaw elementarnych operacji dostępu do plików zwykłych. Właściwa interpretacja zawartości takiego pliku jest sprawą użytkownika lub aplikacji, a informacja o typie pliku, rozpoznawana na poziomie aplikacji, zwyczajowo zawarta jest w nazwie — w jej części po kropce, zwanej rozszerzeniem.



Systemy operacyjne

Struktura pliku

- Struktura logiczna
 - określa organizację informacji wewnątrz pliku
 - może być definiowana i rozpoznawana na poziomie jądra systemu operacyjnego lub na poziomie aplikacji
- Struktura fizyczna
 - określa sposób przechowywania informacji
 - narzucana jest przez urządzenie, na którym plik jest składowany

System plików — warstwa logiczna (7)

Logiczna struktura pliku określa powiązanie informacji wewnątrz pliku (właściwym byłoby zatem określenie struktury informacji). Jako przykład można sobie wyobrazić plik z tabelą bazy danych, w którym:

- pierwsze 4 bajty określają liczbę rekordów (krotek),
- następna 2 długość rekordu w bajtach, kolejny bajt długość nagłówka z definicją atrybutów,
- reszta przeznaczona jest na dane, przy czym każdy rekord ma dodatkowo 2 bajty kontrolne.

Z długości nagłówka wynika, gdzie rozpoczynają się dane, z wielkości rekordu oraz informacji o strukturze, mówiącej o 2 dodatkowych bajtach dla każdego rekordu, można wyliczyć początek rekordu o podanym numerze itd.

Struktura logiczna wiąże się najczęściej z typem pliku, tzn. pewne typy pliku mogą mieć określoną strukturę, np. katalogi w systemie UNIX, plik z obrazem — JPEG lub GIF (Graphics Interchange Format) itd.

Struktura może być definiowana i rozpoznawana na poziomie jądra systemu operacyjnego lub może być rozpoznawana na poziomie aplikacji korzystającej z tego pliku. Definiowanie obsługi różnych struktur plików na poziomie jądra może być pomocne dla użytkownika (może wspomagać optymalizację dostępu), ale w systemie musi być wówczas zawarty kod do obsługi każdej z tych struktur, co może powodować nadmierny rozrost programu jądra.



Fizyczna struktura pliku wynika z własności urządzenia, na którym plik jest przechowywany. Większość urządzeń składowania danych (w tym plików) ma charakter blokowy. Zadaniem systemu operacyjnego jest zatem odwzorować jednostkę logiczną (np. rekord) na jednostkę fizyczną (np. sektor dysku). Operacja dostępu do rekordu może więc wymagać jednej lub kilku operacji dostępu do sektora dysku lub dostępu do bufora pamięci podręcznej, gdzie przechowane są tymczasowe zawartości niektórych obszarów przestrzeni dyskowej.

W związku z blokowym charakterem urządzeń składujących może wystąpić zjawisko fragmentacji wewnętrznej. Wynika ono z faktu, że rozmiar jednostki alokacji na urządzeniu, będący wielokrotnością (w liczbie jakiejś potęgi dwójki: 1, 2, 4, 8, ...) rozmiaru sektora, nie musi dokładnie odpowiadać potrzebom aplikacji.



- Metody dostępu określają sposób identyfikacji odczytywanego lub zapisywanego fragmentu pliku.
- Użycie określonej metody wynika z charakteru przetwarzania oraz struktury pliku.
- Dostępność określonej metody może być ograniczona ze względu na właściwości urządzenia oraz sposób organizacji lub implementacji systemu plików (zwłaszcza atrybutu *lokalizacja*).

Systemy operacyjne



Wyszczególnienie metod dostępu

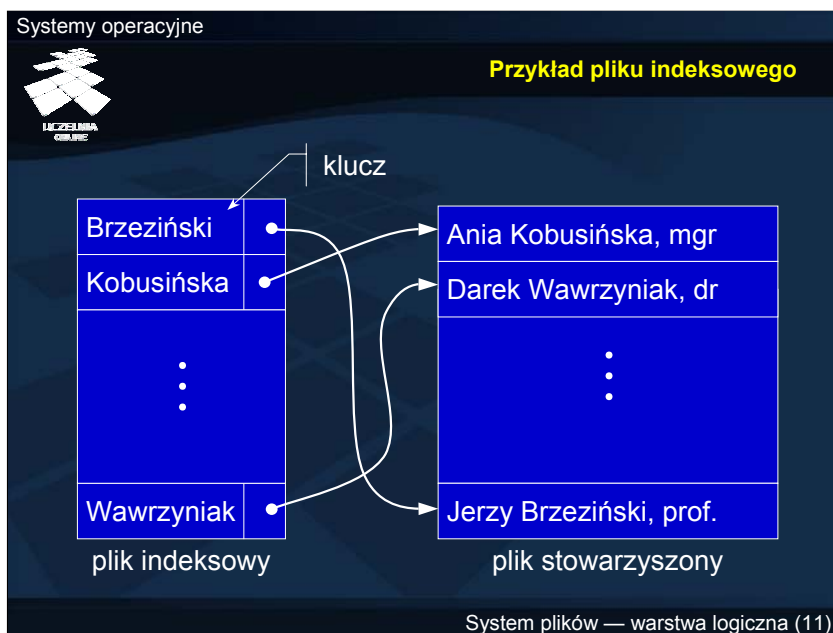
- Dostęp sekwencyjny (ang. sequential access) — informacje w pliku przetwarzane są rekord po rekordzie, tzn. po wykonaniu operacji na określonym rekordzie system przygotowuje się do wykonania operacji na kolejnym rekordzie w pliku.
- Dostęp bezpośredni (swobodny, ang. direct access) — lokalizacja rekordu do przetwarzania podawana jest jako parametr odpowiedniej operacji.
- Dostęp indeksowy — rekord, na którym ma być wykonana operacja identyfikowany jest przez klucz, odwzorowywany na konkretny rekord w pliku stowarzyszonym poprzez plik indeksowy.

System plików — warstwa logiczna (10)

W dostępie sekwencyjnym możliwe są następujące operacje: odczyt kolejnego rekordu, dopisanie rekordu na końcu pliku, przesunięcie wskaźnika bieżącej pozycji (bieżącego rekordu) na początek, ewentualnie przesuwanie wskaźnika o podaną liczbę jednostek w przód, czy w tył. Identyfikacja rekordu dla wykonywanej operacji wynika zatem z historii wcześniejszych operacji dostępu.

W przypadku dostępu bezpośredniego (określanego również jako swobodny) możliwe jest wskazanie dowolnego rekordu (lub innego fragmentu pliku), na którym ma zostać wykonana operacja dostępu. Rekord wskazywany jest poprzez podanie numer lub przesunięcia względem początku pliku. Pliki o dostępie bezpośrednim są szczególnie użyteczne w przetwarzaniu informacji w dużych zbiorach danych (np. w bazach danych).

Dostęp indeksowy umożliwia identyfikowanie rekordów za pomocą kluczy indeksowych. Takie podejścia mają najczęściej na celu przyspieszenie wyszukiwania informacji na podstawie klucza. W praktyce stosowane są raczej w systemach zarządzania bazami danych, niż bezpośrednio w systemie operacyjny. Pliki indeksowe przyjmują często postać skomplikowanych struktur typu B drzewo lub B⁺-drzewo, czy tablice haszowe.



Przykład obrazuje zasadę dostępu indeksowego. Plik indeksowy jest zorganizowany w taki sposób, żeby przyspieszyć wyszukiwanie klucza, np. przez odpowiednie posortowanie lub zbudowanie struktury drzewiastej. Z kluczem związana jest informacja o lokalizacji rekordu w pliku stowarzyszonym, czyli właściwym pliku z danymi.

Sam dostęp polega na znalezieniu na podstawie klucza w pliku indeksowym informacji o lokalizacji rekordu, a następnie wykorzystaniu tej informacji w uzyskaniu dostępu do właściwego rekordu w pliku stowarzyszonym.

Systemy operacyjne



Podstawowe operacje na plikach ⁽¹⁾

- Tworzenie pliku — konieczne jest określenie podstawowych atrybutów pliku, znalezienie miejsca na ten plik w systemie komputerowym oraz jego zaewidencjonowanie (utworzenie wpisu katalogowego)
- Zapis do pliku — konieczne jest określenie, co ma być zapisane i gdzie ma być zapisane (w którym pliku i w jakim miejscu tego pliku, zależnie od sposobu dostępu)
- Odczyt z pliku — konieczne jest określenie, co ma być odczytane (z którego pliku i z jakiego miejsca tego pliku, zależnie od sposobu dostępu) i gdzie mają być umieszczone odczytane dane

System plików — warstwa logiczna (12)

Podstawowe operacje na plikach to tworzenie, usuwanie oraz dostęp do zawartości pliku. Dostęp do zawartości w najprostszym przypadku sprowadza się do zapisu lub odczytu fragmentu pliku.

Tworzenie pliku wymaga jawnego podania wartości niektórych atrybutów — nazwy, typu, czy atrybutów związanych z ochroną. Wartości innych atrybutów definiowane są w trakcie użytkowania pliku, np. czasy dostępu, rozmiar, czy lokalizacja.

W przypadku właściwych operacji dostępu typowy zestaw informacji sprowadza się do określenia „co i gdzie”, czyli co jest zapisywane i w którym miejscu pliku lub która część pliku jest odczytywana oraz gdzie trafia odczytana informacja.

Systemy operacyjne



Podstawowe operacje na plikach (2)

- Usuwanie informacji z pliku — należy określić jaki fragment pliku (i którego pliku) ma być usunięty. Najczęściej możliwe jest tylko skracanie pliku, czyli usuwanie jego końcowej zawartości lub całej jego zawartości.
- Usuwanie pliku — należy określić plik do usunięcia. Usuwana jest zawartość oraz wpis ewidencyjny pliku.
- Dodatkowe operacje na plikach, wykonywane w celu uzyskania dostępu do zawartości pliku:
 - otwieranie,
 - zamykanie,
 - przesuwanie wskaźnika bieżącej pozycji.

System plików — warstwa logiczna (13)

Usuwanie fragmentów pliku może przysparzać problemów. Zasadniczą kwestią jest, co zrobić z miejsce po usuniętej części pliku. W najprostszym przypadku można odpowiednio przesunąć zawartość pliku, ale jest to operacja pracochłonna. W przypadku pliku składającego się z logicznych rekordów o stałym rozmiarze można przenieść ostatni rekord w miejsce usuwanego. Jeśli jednak rekordy są posortowane, metoda ta może się okazać niewłaściwa. Sposób wykonania takiej operacji zależy więc od struktury pliku. Z tego powodu system operacyjny umożliwia tylko usuwanie końcowego fragmentu pliku, wszelkie optymalizacje pozostawiając warstwie aplikacyjnej.

W interfejsie usług plikowych wyróżnia się pewne operacje dodatkowe, jak otwieranie, zamykanie i przesuwanie wskaźnika bieżącej pozycji. Otwieranie oraz komplementarne zamykanie związane jest efektywnością realizacji operacji plikowych. Wygodnym dla użytkownika identyfikatorem pliku jest nazwa. W złożonej, hierarchicznej strukturze katalogów (w tym miejscu odwołujemy się do intuicji słuchacza, gdyż struktury katalogowe zostaną omówione w dalszej części) lokalizowanie pliku na podstawie nazwy wymaga czasami przeszukania kilku katalogów. Identyfikowanie pliku przez nazwę przy każdej operacji dostępu wymagałoby potencjalnie każdorazowego przeszukiwania takiej złożonej struktury katalogowej, co znacząco wydłużyłoby czas dostępu. Otwarcie pliku oznacza alokację odpowiednich zasobów jądra (pozycji w tablicach otwartych plików), które przechowują niezbędne dane od efektywnej realizacji operacji dostępu. Z otwarciem pliku wiąże się utworzenie jakiegoś tymczasowego identyfikatora (deskryptora, uchwytu), za pomocą którego można bardzo szybko zlokalizować plik lub jego fragment.

Sens operacji przesuwania wskaźnika bieżącej pozycji zostanie wyjaśniony w dalszej części.

Systemy operacyjne



Interfejs dostępu do pliku w systemie uniksopodobnym

- Tworzenie pliku — funkcja `creat`
- Usuwanie dowiązania do pliku — funkcja `unlink`
- Odczyt z pliku — funkcja `read`
- Zapis do pliku — funkcja `write`
- Skracanie pliku (usuwanie końcowej zawartości, Obcinanie) — funkcja `truncate`
- Przesunięcie wskaźnika bieżącej pozycji — funkcja `lseek`
- Otwieranie pliku — funkcja `open`
- Zamykanie otwartego pliku — funkcja `close`

System plików — warstwa logiczna (14)

Interfejs dostępu do plików w systemach uniksopodobnych obejmuje operacje wyszczególnione na wcześniejszym slajdzie, jednak kilka funkcji wymaga komentarza.

Funkcja `unlink` usuwa **dowiązanie** do pliku. W systemie uniksopodobnym plik może mieć kilka nazw, zwanych dowiązaniami twardymi. Funkcja `unlink` usuwa wskazane dowiązanie, co nie musi oznaczać fizycznego usunięcia pliku. Fizyczne usunięcie pliku następuje dopiero w wyniku usunięcia ostatniego dowiązania.

Funkcja `creat` (bez litery `e` na końcu), jak nazwa wskazuje, służy do tworzenia pliku. W tym samym celu można jednak wykorzystać funkcję `open`, która również umożliwia utworzenie pliku, ale udostępnia więcej parametrów i tym samym daje większe możliwości w zakresie obsługi przypadków szczególnych.

W opisie interfejsu pominięto między innymi funkcję `fcntl`, oraz kilka funkcji dostępu do specyficznych atrybutów plików.



System uniksopodobny — tworzenie pliku

- Funkcja systemowa: `creat`
- Parametry:
 - nazwa pliku (ścieżka)
 - prawa dostępu
- Wartość zwrótna: deskryptor pliku lub -1 w przypadku błędu.
- Uwaga: funkcja tworzy plik i otwiera go do zapisu.



- Funkcja systemowa: `open`
- Parametry:
 - nazwa pliku (ścieżka)
 - tryb otwarcia (do zapisu, do odczytu itp.)
- Wartość zwrótna: deskryptor pliku lub -1 w przypadku błędu.
- Wersja 3-argumentowa funkcji umożliwia również tworzenie pliku.



System uniksopodobny — zamykanie deskryptora pliku

- Funkcja systemowa: `close`
- Parametry:
 - deskryptor
- Wartość zwrótna: 0 w przypadku poprawnego zakończenia lub -1 w przypadku błędu.



System uniksopodobny — usuwanie dowiązania do pliku

- Funkcja systemowa: `unlink`
- Parametry:
 - nazwa pliku (ścieżka)
- Wartość zwrótna: 0 w przypadku poprawnego zakończenia lub -1 w przypadku błędu.

**System uniksopodobny — skracanie pliku**

- Funkcja systemowa: `truncate`
- Parametry:
 - nazwa pliku (ścieżka) lub deskryptor
 - wielkość w bajtach do jakiej ma nastąpić skrócenie (docelowa wielkość pliku)
- Wartość zwrotna: 0 w przypadku poprawnego zakończenia lub -1 w przypadku błędu.

Systemy operacyjne



System uniksopodobny — odczyt zawartości pliku

- Funkcja systemowa: `read`
- Parametry:
 - deskryptor pliku
 - adres w pamięci, pod którym zostaną zapisane dane odczytane z pliku
 - liczba odczytywanych bajtów
- Wartość zwrotna: liczba rzeczywiście odczytanych bajtów lub -1 w przypadku błędu.

System plików — warstwa logiczna (20)

W parametrach funkcji `read` nie ma wskazania na odczytywany fragment pliku. Takie podejście jest właściwe dla dostępu sekwencyjnego.

Systemy operacyjne

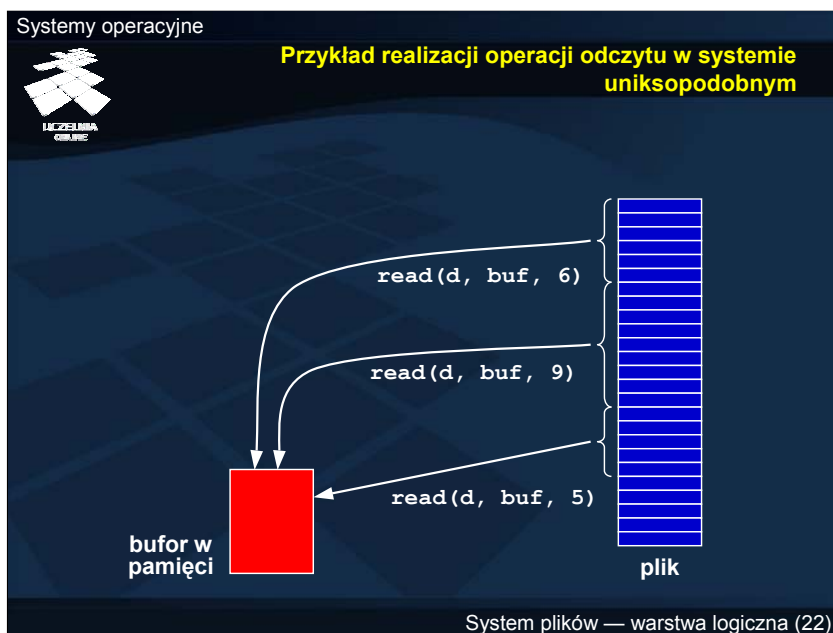


System uniksopodobny — zapis zawartości pliku

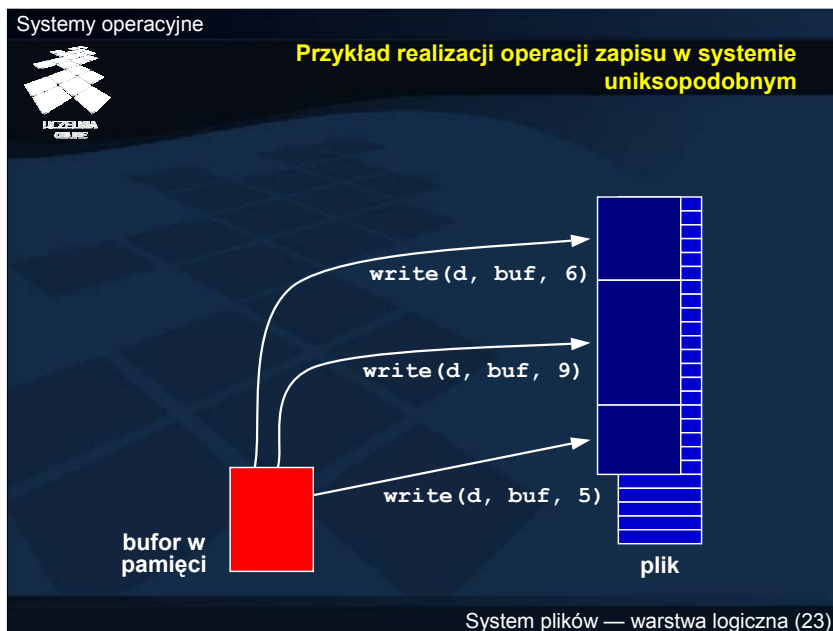
- Funkcja systemowa: `write`
- Parametry:
 - deskryptor pliku
 - adres w pamięci, spod którego zostaną pobrane dane do zapisu
 - liczba zapisywanych bajtów
- Wartość zwrotna: liczba rzeczywiście zapisanych bajtów lub -1 w przypadku błędu.

System plików — warstwa logiczna (21)

Podobnie jak w przypadku funkcji `read`, w parametrach funkcji `write` również nie ma wskazania, który obszar pliku ma być zapisywany. Wpływ na miejsce zapisywania może mieć tryb otwarcia. W przypadku otwarcia pliku z flagą `O_APPEND` następuje zawsze dopisywanie na końcu. Jeśli flaga nie została użyta, w wyniku wykonania funkcji może nastąpić nadpisanie dotychczasowej zawartości, przy czym plik zachowuje się tak, jak przy dostępie sekwencyjnym.



Przykład pokazuje efekt trzech kolejnych operacji odczytu pliku po jego otwarciu. Przyjmując od góry początek pliku, pierwszy wykonanie `read` umieści w buforze pamięci pierwsze sześć bajtów, następne nadpisze te sześć bajtów kolejnymi dziewięcioma z pliku, ostatnie wykonanie częściowo nadpisze zawartość bufora kolejnymi pięcioma bajtami. Jest to typowy przykład dostępu sekwencyjnego. Po wykonaniu operacji stan otwarcia pliku zmienia się w tak, że następna operacja dotyczy kolejnego fragmentu pliku.



Podobnie realizowana jest operacja zapisu. Wynikiem kolejnych trzech wywołań funkcji `write` będzie zapisanie łącznie 20 kolejnych bajtów w pliku. Jeśli w zapisywanym obszarze pliku były już jakieś dane, zostaną nadpisane. Jeśli plik byłby krótszy niż 20 bajtów, nastąpi jego powiększenie.



Systemy operacyjne

System uniksopodobny — zmiana wskazania bieżącej pozycji

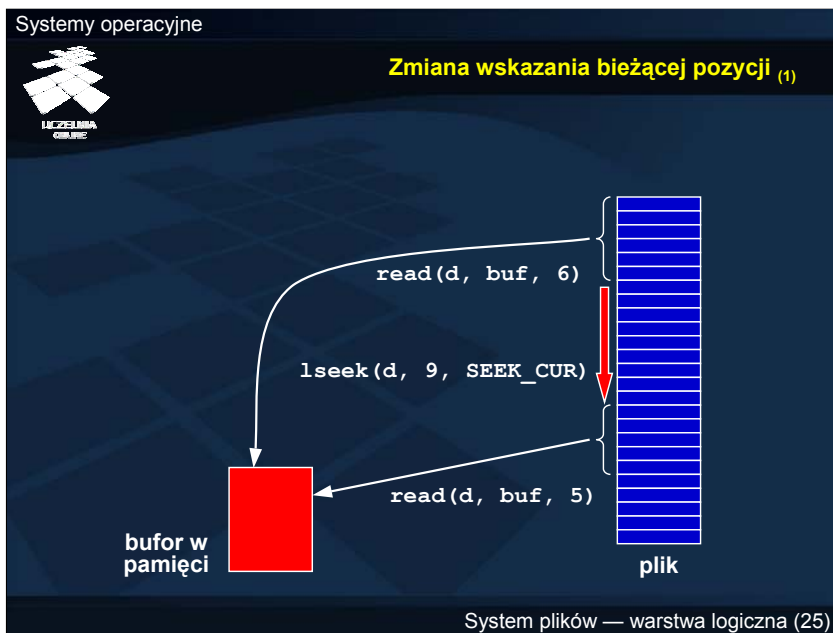
- Funkcja systemowa: `lseek`
- Parametry:
 - deskryptor pliku
 - wielkość przesunięcia w bajtach
 - punkt odniesienia (początek pliku, bieżąca pozycja, lub koniec pliku)
- Wartość zwrotna: położenie wskaźnika po przesunięciu lub -1 w przypadku błędu.

System plików — warstwa logiczna (24)

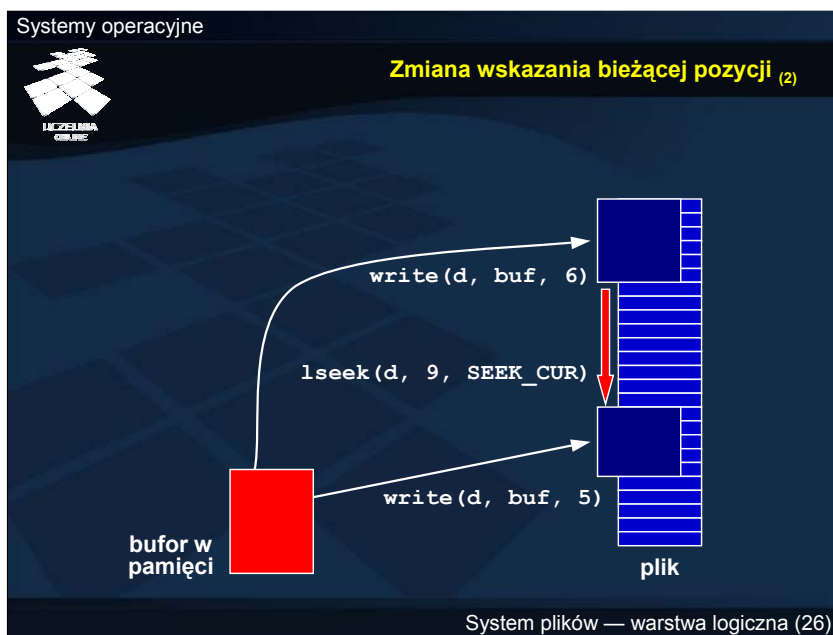
Z każdym otwartym plikiem związany jest wskaźnik bieżącej pozycji. Wskaźnik ten identyfikuje miejsce, od którego rozpocznie się kolejna operacja zapisu lub odczytu. Po każdej operacji wskaźnik bieżącej pozycji przesuwa się o tyle bajtów, na ile operacja była wykonana.

Niezależnie od wykonywanych operacji dostępu wskaźnik ten można przesunąć za pomocą funkcji `lseek`. Wskazanie miejsca docelowego polega na podaniu przesunięcia względem początku pliku, końca pliku lub pozycji bieżącej. Wartość dodatnia oznacza przesunięcie w kierunku końca pliku, a ujemna w kierunku początku pliku.

Funkcja `lseek` jest uzupełnieniem funkcji `read` i `write`, umożliwiającym bezpośredni dostęp do plików. W przypadkach plików sekwencyjnych, do których należą np. pliki urządzeń, czy łącza (potoki, kolejki FIFO) funkcja `lseek` nie ma zastosowania. W ten sposób udało się zachować jednolitość interfejsu dostępu do plików — nie trzeba było wyodrębniać dwóch rodzajów funkcji do odczytu i do zapisu.



Przykład pokazuje przykładową realizację operacji odczytu z przesunięciem wskaźnika bieżącej pozycji o 9 bajtów w kierunku końca pliku. Skutkiem jest pominięcie tych 9 bajtów.



W przypadku zapisu skutkiem przesunięcia jest pominięcie 9 bajtów, czyli pozostawienie tego fragmentu bez zmian.

Systemy operacyjne



Organizacja logiczna systemu plików ⁽¹⁾

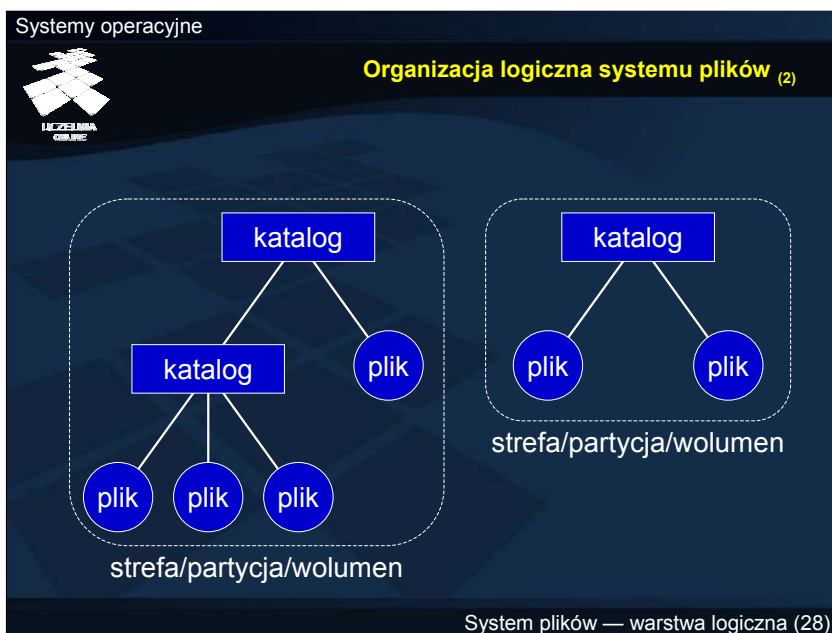
- Podział na strefy (wolumeny, woluminy, tomy, partycje)
 - strefa obejmuje część dysku, jeden lub kilka dysków,
 - strefa zawiera pliki i katalogi.
- Organizacja katalogów:
 - katalog jest tablicą kojarzącą nazwy plików z wpisami katalogowymi, obejmującymi inne atrybuty plików,
 - katalogi mogą być jedno- lub wielopoziomowe,
 - katalogi wielopoziomowe zorganizowane mogą być w różne struktury logiczne (drzewo, graf acykliczny, dowolny graf).
- Pliki identyfikowane są przez nazwy, znajdujące się w katalogach.

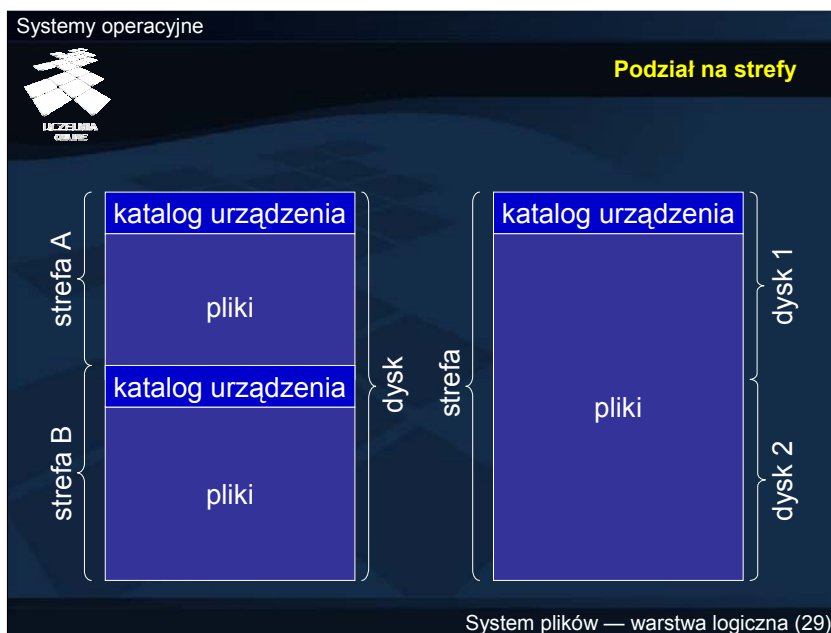
System plików — warstwa logiczna (27)

System plików daje abstrakcyjny obraz informacji, gromadzonej w systemie komputerowym. Fizycznie, informacja ta może być przechowywana na różnych nośnikach: taśmach, dyskach, płytach kompaktowych, a nawet w pamięci operacyjnej. Strefa jest najczęściej obszarem związanym z fizycznym urządzeniem przechowującym plik. Czasami na urządzeniu (np. dysku) wydziela się różne strefy, postrzegane jako logiczne urządzenia.

Pliki w strefie znajdują się w katalogach, które z kolei najczęściej tworzą strukturę hierarchiczną. Ze daną strefą związana jest na ogół zbiór danych o odpowiedniej strukturze do identyfikacji hierarchii katalogów oraz lokalizacji bloków z zawartością plików. Informacje takie określa się jako *metadane*. Inicjalizacja metadanych nazywana jest *formatowaniem logicznym*, albo tworzeniem systemu plików.

Warto podkreślić, że jest to logiczny obraz, który może się różnie przekładać na fizyczne rozlokowanie danych na dostępnych urządzeniach.





Jak już wspomniano, strefa odpowiada najczęściej urządzeniu fizycznemu. Podział na strefy może być też jednak zrobiony w celu wyodrębnienia kilku logicznych urządzeń (dysków) na jednym urządzeniu fizycznym lub powiązania kilku fizycznych urządzeń w jedną logiczną całość.

Systemy operacyjne



Operacje na katalogu

- Tworzenie katalogu
- Usuwanie katalogu
- Tworzenie wpisu katalogowego — gdy tworzony jest plik, jego nazwa alternatywna, podkatalog itp.
- Usuwanie wpisu katalogowego
- Przemianowanie pliku (zmiana nazwy)
- Odnajdowanie wpisu katalogowego
- Tworzenie wykazu wpisów katalogowych (listing zawartości)

System plików — warstwa logiczna (30)

Wpis w katalogu tworzony jest zawsze, gdy jakiś obiekt z danymi ma być identyfikowany przez nazwę za pośrednictwem systemu plików. Obiektem takim jest plik. Plik może mieć kilka alternatywnych nazw, z których każda ma swój wpis. Swoje wpisy mają też podkatalogi w przypadku struktury hierarchicznej. Wpis taki można również usunąć lub przemianować. Wszystkie te operacje dotyczą modyfikacji katalogu. Katalog na ogół częściej jest przeszukiwany niż modyfikowany. Przeszukiwanie polega na poszukiwaniu wpisu, identyfikowanego przez nazwę lub tworzeniu wykazu wszystkich wpisów, spełniających określone kryteria.

Systemy operacyjne

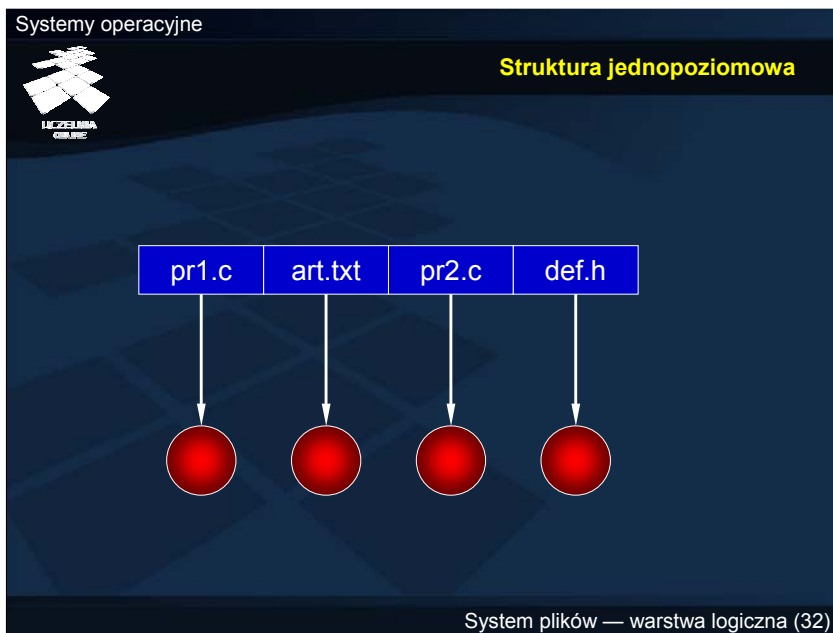


Struktura logiczna katalogów

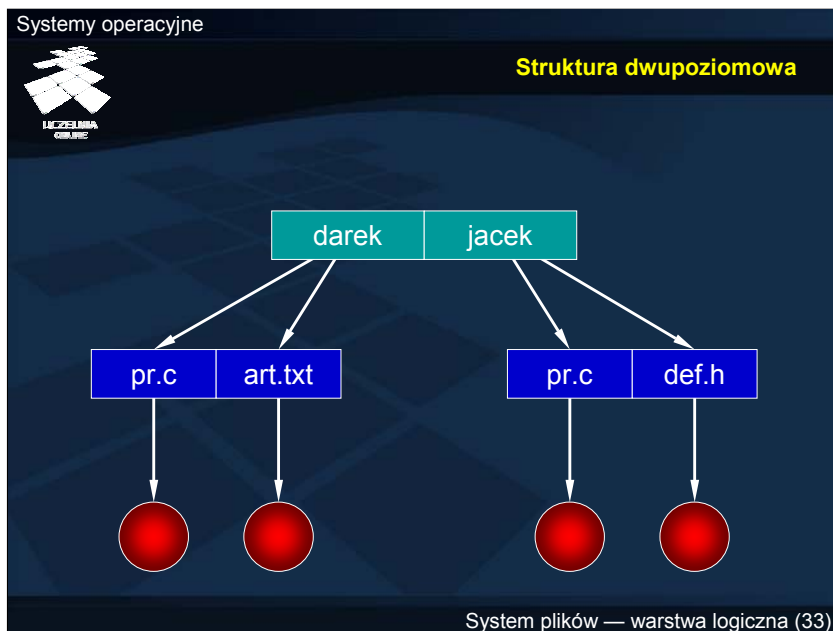
- Struktura jednopoziomowa — wpisy katalogowe poszczególnych plików znajdują się w tym samym katalogu (na tym samym poziomie).
- Struktura dwupoziomowa — wpisy katalogowe plików znajdują się w różnych katalogach, ale katalogi nie mogą zawierać innych katalogów.
- Struktura drzewiasta — w katalogach można tworzyć podkatalogi oraz pliki.
- Graf acykliczny — podkatalog (lub plik) może być umieszczony w wielu katalogach.
- Graf ogólny — dopuszcza się cykl w powiązaniach pomiędzy katalogami

System plików — warstwa logiczna (31)

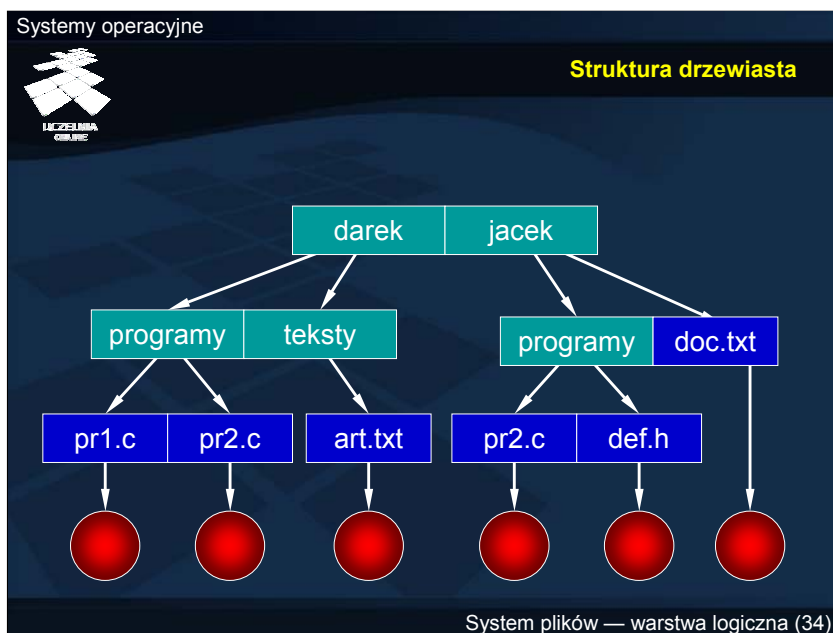
Współcześnie najczęściej katalogi są logicznie powiązane w strukturę drzewa, która narzuca naturalną hierarchiczność. W przeszłości stosowane był niekiedy prostsze struktury, np.. jednopoziomowa, dwupoziomowa. Można sobie również wyobrazić bardziej skomplikowane struktury, np. graf acykliczny, czy graf dowolny.



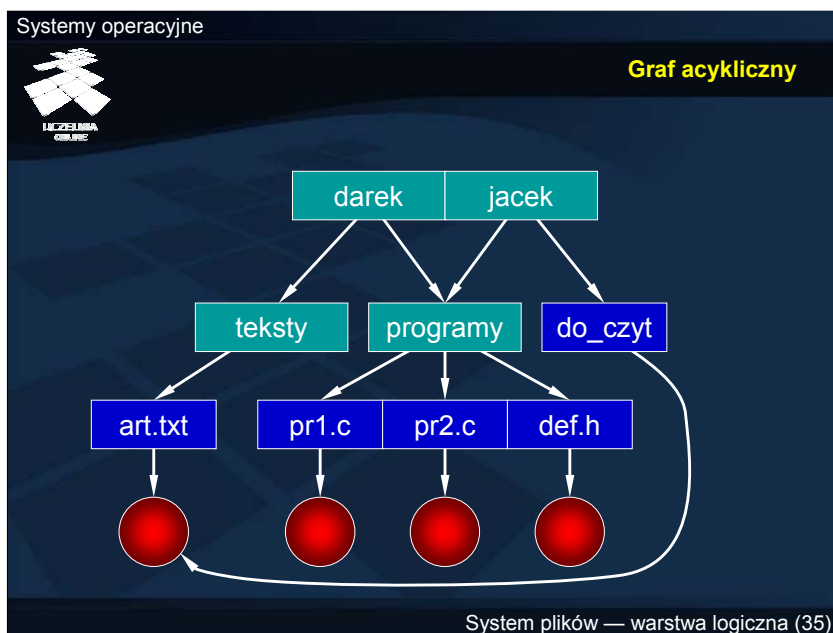
Wszystkie pliki identyfikowane są przez nazwy, znajdujące się na tym samym poziomie. Wymaga się zatem, żeby nazwy te były unikalne. Jest to poważne ograniczenie zwłaszcza w systemach wielodostępnych.



Krokiem w kierunku ułatwienia wielodostępu przy „płaskiej” strukturze katalogów jest struktura dwupoziomowa. Każdy użytkownik ma swój katalog, więc wybierane przez niego nazwy, nie kolidują z nazwami, wybranymi przez innych użytkowników. Podejście takie zastosowano w systemie CP/M.

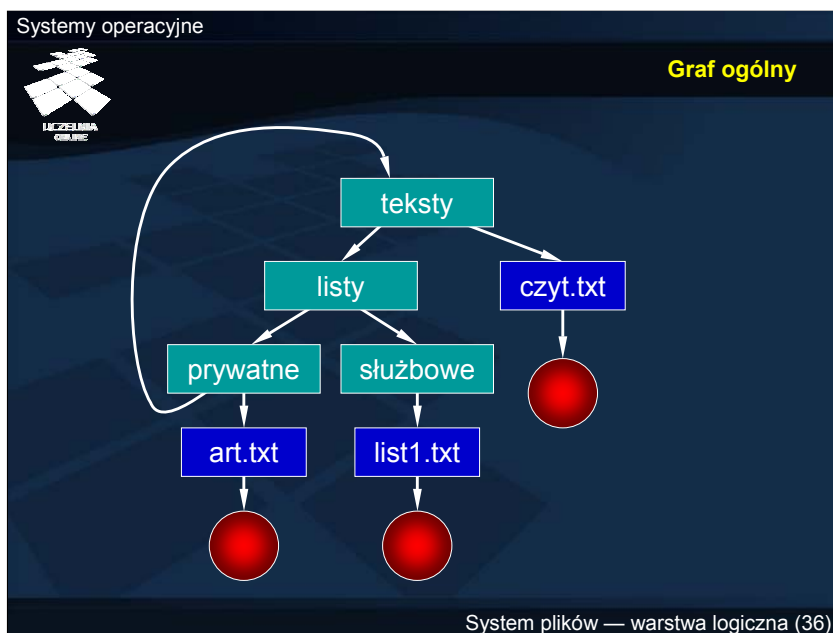


Typowym podejściem jest organizacja katalogów w formie drzewa. Każdy katalog zawiera wpisy odpowiadające plikom oraz innym katalogom. Nazwy w każdym katalogu muszą być unikalne, mogą się natomiast powtarzać w innych katalogach.



Graf acykliczny oznacza, że ten sam obiekt może mieć wpis (dowiązanie) w kilku katalogach. W prezentowanym przykładzie obiekt (katalog) *programy* ma wpis w nadkatalogu *darek* oraz nadkatalogu *jack*. Podobnie jeden z plików ma dwie różne nazwy — *art.txt* i *do_czyt* — w różnych katalogach.

Tego typu organizacja logiczna plików i katalogów stosowana jest w systemach uniksopodobnych, ale wielokrotne dowiązanie nie jest możliwe dla katalogów. Poza tym dowiązanie te muszą być w tej samej partycji (logicznym urządzeniu).



Dowolny graf raczej nie jest stosowany w systemach plików, gdyż dopuszcza cykl, który utrudnia orientację w trakcie poruszania się po takiej strukturze. Hierarchia jest najlepszą formą organizacji informacji, akceptowalną przez ludzki mózg i tym samym wygodną dla użytkownika.