

KOMPONENTY SYSTEMÓW OPERACYJNYCH

dr hab. inż. Krzysztof Patan, prof. PWSZ

Instytut Politechniczny
Państwowa Wyższa Szkoła Zawodowa w Głogowie
`k.patan@issi.uz.zgora.pl`

PRZYDZIAŁ CZASU PROCESORA

Cel:

Stałe wykorzystanie jednostki centralnej poprzez utrzymywanie w działaniu pewnej liczby zadań

- Wieloprogramowanie:
 - ◇ proces jest wykonywany do momentu, aż będzie musiał czekać do przeprowadzenia operacji wejścia–wyjścia
 - ◇ system operacyjny przydziela wtedy procesorowi kolejne zadanie do wykonania
 - ◇ zawsze gdy któryś z procesów musi czekać inny jest wykonywany

Kryteria planowania

- Różne algorytmy planowania przydziału procesora mają rozmaite właściwości i do pewnych klas procesów mogą nadawać się lepiej niż do innych
- Do porównania algorytmów planowania stosuje się wiele kryteriów oceny
- Najczęściej stosowane kryteria to:
 - 1 **wykorzystanie procesora** – procesor powinien być nieustannie zajęty pracą. Wykorzystanie procesora może zawierać się od 0 do 100%, np.:
 - 40% – słabe wykorzystanie procesora
 - 90% – intensywna eksploatacja systemu
 - 2 **przepustowość** – liczba procesów kończonych w jednostce czasu
 - dla długich procesów – 1/h
 - dla krótkich procesów – 10/sek

- ③ **czas cyklu przetwarzania** – czas potrzebny na wykonanie procesu. Jest to czas upływający między chwilą nadejścia procesu do systemu a chwilą jego zakończenia (czas obiegu zadania)
 - ④ **czas oczekiwania** – algorytm planowania nie ma wpływu na czas, w którym proces wykonuje operacje IO. Kryterium to uwzględnia tylko czasu, który proces spędza w kolejce procesów gotowych do wykonania
 - ⑤ **czas odpowiedzi** – czas cyklu przetwarzania nie jest najlepszym kryterium dla systemów interakcyjnych. To kryterium uwzględnia czas upływający między wysłaniem żądania, a pojawieniem się pierwszej odpowiedzi
- przepustowość i wykorzystanie procesora – maksymalne
 - czas cyklu przetwarzania, czas oczekiwania oraz czas odpowiedzi – minimalne
 - często optymalizuje się wartość średnią

Algorytmy planowania

Planowanie przydziału procesora jest związane z zagadnieniami podejmowania decyzji o tym, którym procesom z kolejki procesów gotowych do działania ma być przydzielony procesor

Metoda FCFS

- FCFS (ang. *First Come First Served*) – pierwszy zgłoszony pierwszy obsłużony
- proces który pierwszy zamówi procesor pierwszy go otrzyma
- metodę można zaimplementować przy pomocy kolejki FIFO (ang. *First In First Out*)
- Proces wchodzący do kolejki jest dołączany na samym końcu kolejki
- metoda bardzo prosta do zrozumienia i implementacji

Przykład 1.

Kolejkę procesów gotowych do wykonania zawiera poniższa tabela. Kolejność nadchodzenia procesów P_1 , P_2 , P_3 . W celu uproszczenia rozważamy po jednej fazie procesora dla każdego procesu.

Proces	Czas trwania fazy procesora [ms]
P_1	24
P_2	3
P_3	3

- realizacja procesów z kolejki



- średni czas oczekiwania

$$t_{sr} = \frac{0 + 24 + 27}{3} = 17ms$$

- dla procesów nadchodzących w kolejności P_2 , P_3 , P_1



$$t_{sr} = \frac{0 + 3 + 6}{3} = 3ms$$

Wnioski

- kolejność wykonywania procesów może mieć fundamentalne znaczenie na wartość średniego czasu oczekiwania
- średni czas oczekiwania w metodzie FCFS nie jest optymalny
- ograniczenia metody są widoczne wtedy gdy mamy jeden proces ograniczony przez procesor i kilka procesów ograniczonych przez urządzenia wejścia–wyjścia
- efekt konwoju – wszystkie procesy w kolejce czekają na zwolnienie procesora przez jeden długi proces
- metoda FCFS jest szczególnie kłopotliwa w systemach z podziałem czasu

Planowanie priorytetowe

- każdemu procesowi przypisuje się priorytet
- procesor przydziela się temu procesowi, który posiada najwyższy priorytet
- procesy o równych priorytetach planuje się metodą FCFS
- na algorytm SJF można spojrzeć jak na algorytm priorytetowy, w którym priorytet jest odwrotnością długości fazy procesora
- priorytety zwykle określa się liczbami całkowitymi, np. od 0 do 7, przy czym 0 może oznaczać najniższy priorytet

Przykład 2.

Rozważmy pulę zadań gotowych do wykonania.

Proces	Czas trwania fazy procesora [ms]	Priorytet
P_1	10	3
P_2	1	1
P_3	2	3
P_4	1	4
P_5	5	2

- realizacja procesów z kolejki



- średni czas oczekiwania

$$t_{sr} = \frac{0 + 1 + 11 + 13 + 18}{5} = 8.6ms$$

Wnioski

- priorytety mogą być definiowane wewnętrznie – za pomocą mierzalnej właściwości procesu (limit czasu, liczba otwartych plików, itp.)
- priorytety mogą być definiowane za pomocą kryteriów zewnętrznych – ważność procesu, kwota opłat ponoszonych za użytkowanie komputera, itp.
- poważny problem stanowi zagłódzenie procesu
- rozwiązanie problemu zagłódzenia – postarzanie procesu

Planowanie rotacyjne

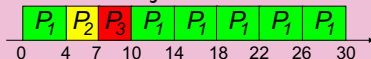
- algorytm zaprojektowano dla systemów z podziałem czasu
- jest to wywłaszczeniowy algorytm FCFS
- przełączanie procesów następuje z czasem zwanym kwantem lub odcinkiem czasu (10 – 100ms)
- każdemu procesowi przydziela się procesor na czas nie dłuższy od kwantu czasu
- procesy do wykonania pobierane są z kolejki procesów gotowych do wykonania metodą FCFS
- jeśli faza procesora procesu wykonywanego jest krótsza niż 1 kwant czasu, wówczas proces z własnej inicjatywy zwolni procesor

Przykład 3.

Rozważmy pulę zadań gotowych do wykonania. Rozplanować je metodą rotacyjną z kwantem = 4ms

Proces	Czas trwania fazy procesora [ms]
P_1	24
P_2	3
P_3	3

- realizacja procesów z kolejki



- średni czas oczekiwania

$$t_{sr} = \frac{(10 - 4) + 4 + 7}{3} = 5.66ms$$

Wnioski

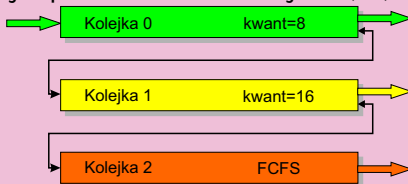
- wadą tej metody jest długi średni czas oczekiwania
- wydajność metody w dużym stopniu zależy od rozmiaru kwantu czasu
 - ❶ kwant czasu bardzo długi – metoda FCFS
 - ❷ kwant czasu bardzo mały – dzielenie procesora

Wielopoziomowe planowanie kolejek ze sprzężeniem zwrotnym

- metoda jest rozwinięciem wielopoziomowego planowania kolejek, umożliwienie przemieszczenia się procesów pomiędzy kolejkami
- rozdziela się procesy o różnych długościach faz procesora i umieszcza w różnych kolejkach
- proces zużywający za dużo czasu procesora zostaje przeniesiony do kolejki o niższym priorytecie wykonania
- proces oczekujący zbyt długo na wykonanie zostanie przeniesiony do kolejki o wyższym priorytecie

Przykład 4.

Rozważmy 3 kolejki ponumerowane kolejno 0, 1, 2.



- 1 planista wykonuje procesy z kolejki 0
- 2 po opróżnieniu kolejki są wykonywane procesy z kolejki 1
- 3 procesy z kolejki 2 są wykonywane dopiero po opróżnieniu kolejki 1
- 4 proces nadchodzący do kolejki 1 wywłączy wykonywany proces z kolejki 2
- 5 proces nadchodzący do kolejki 0 wywłączy wykonywany proces z kolejki 1 lub 2

PRZYDZIAŁ PAMIĘCI OPERACYJNEJ

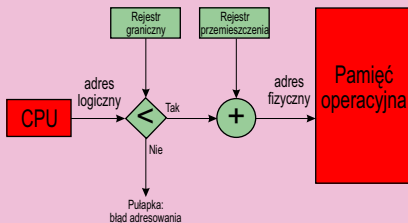
- Pamięć komputera – wielka tablica słów (bajtów) oznaczonych adresami
- Wykonanie programu:
 - ◇ proces jest ładowany do pamięci
 - ◇ procesor pobiera rozkazy z pamięci stosownie do wartości licznika rozkazów
- Typowy cykl wykonania rozkazu
 - ◇ pobranie rozkazu z pamięci
 - ◇ dekodowanie rozkazu
 - ◇ pobranie z pamięci dodatkowych argumentów
 - ◇ wyniki są przechowywane w pamięci

Przydział ciągły pamięci operacyjnej

Założmy, że system operacyjny znajduje się w pamięci dolnej, a procesy użytkownika w górnej

Zadania algorytmu przydziału pamięci:

- ochrona kodu i danych systemu operacyjnego przed zmianami pochodzącymi od procesów użytkownika
- ochrona procesów użytkownika przed zmianami spowodowanymi przez inne procesy użytkownika



- **Rejestr przemieszczenia** – wartość najmniejszego adresu fizycznego
- **Rejestr graniczny** – zakres adresów logicznych
- Planista CPU wybiera proces do wykonania, wtedy ekspedytor nadaje odpowiednie wartości rejestrom

Przydział wielu obszarów

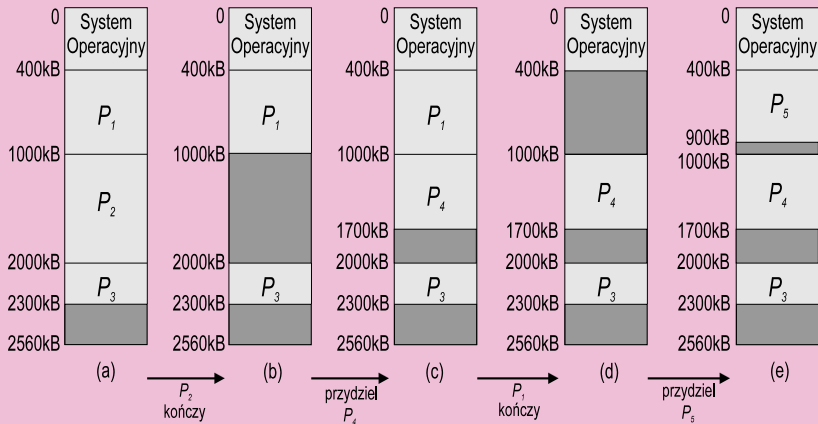
- Maksymalne wykorzystanie pamięci – wiele procesów użytkowych może być załadowanych do pamięci w tym samym czasie
- Cała dostępna pamięć traktowana jest jak jeden wielki blok – dziura
- Dla procesu z zapotrzebowaniem na pamięć szukana jest odpowiednio duża dziura

Przykład

Mamy pamięć o rozmiarze 2560 kB i system operacyjny zajmujący 400kB. Rozplanować pamięć biorąc pod uwagę kolejkę z tabeli

Proces	Pamięć	Faza procesora
P_1	600kB	10
P_2	1000kB	5
P_3	300kB	20
P_4	700kB	8
P_5	500kB	15

- planowanie zadań - metoda FCFS
- przydział CPU - metoda rotacyjna (kwant=1)



Wybór wolnych obszarów: strategie pierwszego bądź najlepszego dopasowania

Fragmentacja zewnętrzna

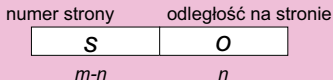
- Przestrzeń wolnej pamięci zostaje podzielona na małe kawałki. Suma wolnych obszarów może wystarczyć na spełnienie pewnego zamówienia, lecz nie tworzą one spójnego obszaru
- Badania statystyczne pokazały, że w strategii pierwszego dopasowania na N przydzielonych bloków z powodu fragmentacji będzie ginąć $0.5N$ innych bloków (1/3 pamięci bezużyteczna! – reguła 50%)

Stronicowanie

Stronicowanie (ang. *paging*) jest metodą pozwalającą na rozwiązanie problemu zewnętrznej fragmentacji. Pozwala się na to, aby procesowi można było przydzielać dowolne dostępne miejsca w pamięci fizycznej

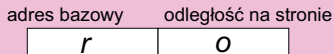
Metoda podstawowa

- Pamięć fizyczną dzieli się na bloki o stałej długości (ramki)
- Pamięć logiczną dzieli się na bloki o takiej samej długości (strony)
- Adres logiczny składa się z dwóch części:
 - 1 numer strony (ang. *page address*) – s
 - 2 odległość na stronie (ang. *page offset*) – o

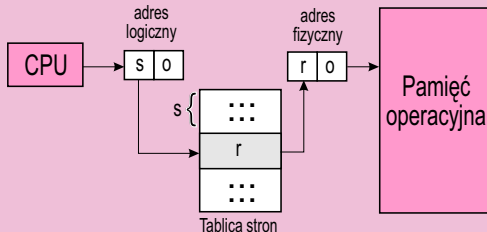


Definicja adresu fizycznego

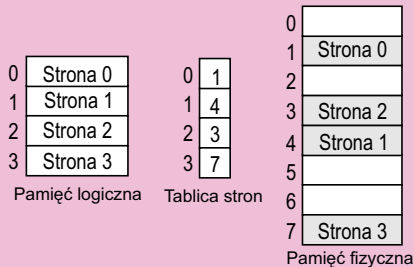
- adres bazowy strony w pamięci – r
- odległość na stronie – o



Idea stronicowania pamięci



- numer strony jest indeksem w tablicy stron
- tablica stron zawiera adresy bazowe wszystkich stron w pamięci



- rozmiar strony (zarazem ramki) określa sprzęt
- rozmiar ten jest zwykle potęgą dwójki:
 - 2^m – rozmiar przestrzeni logicznej
 - 2^n – rozmiar strony

Podsumowanie

- stronicowanie jest odmianą dynamicznego przemieszczenia. Przypomina używanie tablicy rejestrów bazowych - po jednym na każdą ramkę pamięci
- stronicowanie eliminuje zewnętrzną fragmentację
- stronicowanie wprowadza wewnętrzną fragmentację
np. rozmiar strony – 2048B, rozmiar procesu – 72766B
wewnętrzna fragmentacja – 962B
- wyraźne rozdzielenie pamięci widzianej przez użytkownika od pamięci fizycznej
- stronicowanie wydłuża czas przełączania kontekstu.
Przechowuje kopie tablicy stron, licznika rozkazów i zawartości rejestrów

SYSTEM PLIKÓW

System plików

System plików jest tym komponentem systemu operacyjnego, który dostarcza mechanizmy bezpośredniego przechowywania i dostępu do danych oraz do programów

- System plików na stałe rezyduje w pamięci pomocniczej
- W systemie plików można wyodrębnić:
 - ◇ zbiór plików
każdy plik zawiera powiązane ze sobą informacje
 - ◇ strukturę katalogów
organizacja plików na dysku oraz udostępnianie informacji o wszystkich plikach w systemie
 - ◇ strefy (partycje)
wyodrębnianie fizyczne lub logiczne wielkich zbiorów katalogów

Pliki

- ☛ Komputery przechowują informacje na wielu różnych nośnikach: dyski magnetyczne, taśmy magnetyczne, dyski optyczne, etc.
- ☛ System operacyjny powinien dostarczyć jednolity logiczny obraz przechowywanych informacji
- ☛ Niezależnie od fizycznych nośników, system operacyjny definiuje logiczną jednostkę magazynowania informacji – plik

Plik jest zbiorem powiązanych ze sobą informacji, zapisanym w pamięci pomocniczej

- ☛ Niesprecyzowane pojęcie pliku
 - ◇ najczęściej pliki reprezentują programy oraz dane
 - ◇ mogą mieć postać liczbową, alfanumeryczną lub binarną
 - ◇ mogą mieć format swobodny lub ściśle określony
 - ◇ plik może być ciągiem bitów, wierszy lub rekordów

Atrybuty pliku

Dla wygody użytkowników komputera pliki zaopatrywane są w pewne atrybuty (cechy). Zawierają one kluczowe informacje o pliku oraz umożliwiają proste odwoływanie się do niego

- ◇ **Nazwa**

- ◇ **Typ**

informacja o typie pliku potrzebna jest w systemach które rozróżniają typy plików

- ◇ **Położenie**

wskaźnik do urządzenia dyskowego i położenia pliku na tym urządzeniu

- ◇ **Rozmiar**

bieżący rozmiar pliku; czasami maksymalny dopuszczalny rozmiar pliku

- ◇ **Ochrona**

info o tym kto może plik czytać, zapisywać, wykonywać, etc.

- ◇ **Czas, data, identyfikator użytkownika**

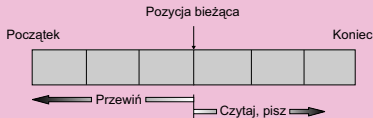
info o czasie utworzenia pliku, ostatniej modyfikacji, ostatnim użyciu pliku

Metody dostępu do pliku

W przypadku potrzeby przeczytania informacji zapisanej w pliku, należy uzyskać do niej dostęp, a następnie przekopiować do pamięci operacyjnej

Dostęp sekwencyjny

- metoda opiera się na taśmowym modelu pliku
- informacje są zapisywane i przetwarzane po kolei, jeden rekord za drugim
- operacja czytania pobiera rekord i przesuwa wskaźnik położenia o jedną pozycję do przodu
- operacja pisania umieszcza dane na końcu pliku i umieszcza wskaźnik położenia za nowo zapisanymi danymi
- możliwość przewijania pliku wstecz lub wprzód o n rekordów

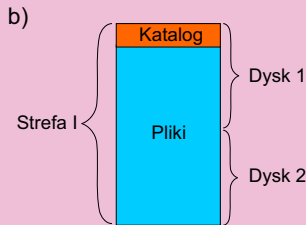
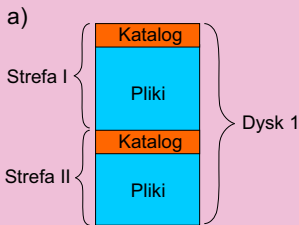


Dostęp bezpośredni (względny)

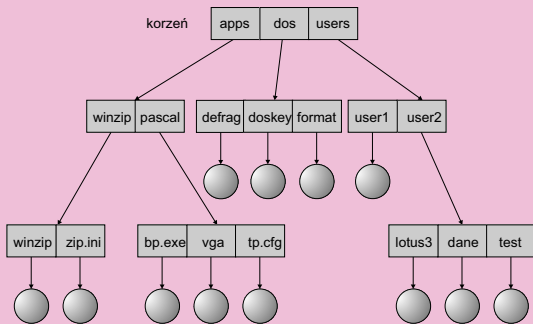
- metoda opera się na dyskowym modelu pliku
- plik składa się z rekordów logicznych o stałej długości
- rekordy mogą być bezpośrednio czytane i zapisywane bez zachowania specjalnego porządku
- operacje plikowe muszą jako parametr zawierać numer rekordu
np. `pisz n`, `czytaj n`
- możliwość symulacji pliku o dostępie sekwencyjnym przez plik o dostępie bezpośrednim

Struktura katalogów

- ☛ System plików podzielony jest na strefy (minidyski, woluminy, wolumeny, partycje)
- ☛ Strefy mieszczą pliki i katalogi
- ☛ Każda strefa zawiera informacje o zawartej w nich plikach i katalogach (katalog urządzenia, tablica zawartości tomu, **katalog**)



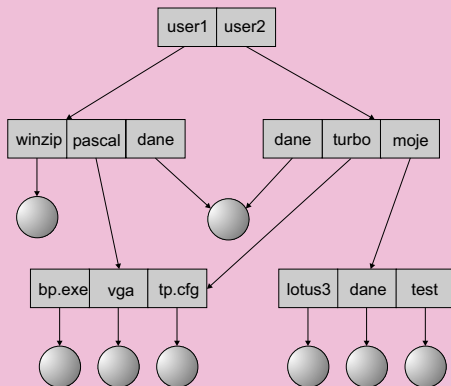
Drzewo katalogów



- Uogólnienie katalogu dwupoziomowego
- Możliwość stosowania przez użytkowników własnych podkatalogów

- Katalog główny – korzeń, pliki – liście
- Katalog bieżący – katalog aktywny, roboczy
- Każdy plik ma jednoznaczną ścieżkę bezwzględną
- Ścieżki względne i bezwzględne
- Ścieżki przeszukiwania – ciąg katalogów przeszukiwanych w celu odnalezienia pliku

Grafi katalogów



- Graf - uogólnienie drzewa katalogów
- Możliwość dzielenia katalogów i plików pomiędzy wielu użytkowników

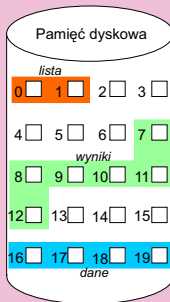
Metody przydziału miejsca na dysku

Problem:

Jak przydzielić miejsce na dysku tak aby obszar dyskowy był zagospodarowany efektywnie, a dostęp do plików szybki?

Przydział ciągły

- plik zajmuje ciąg kolejnych bloków na dysku



Katalog

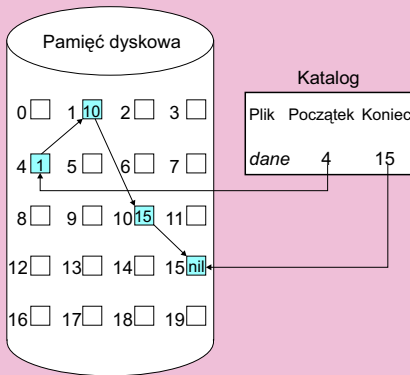
Plik	Początek	Długość
<i>lista</i>	0	2
<i>wyniki</i>	7	6
<i>dane</i>	16	4

Wady

- fragmentacja zewnętrzna
- znajomość wielkości obszaru potrzebnego dla danego pliku (fragmentacja wewnętrzna)

Przydział listowy

- plik jest listą powiązanych ze sobą bloków dyskowych; katalog zawiera wskaźniki do pierwszego i ostatniego bloku pliku
- każdy blok zawiera wskaźnik do kolejnego bloku



Zalety:

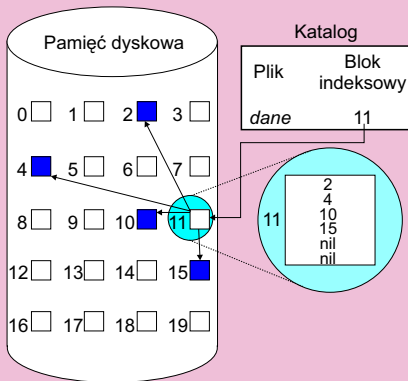
- uniknięcie wad przydziału ciągłego

Wady:

- możliwość stosowania jedynie do pliku o dostępie sekwencyjnym
- wskaźniki zajmują dodatkową przestrzeń
- niska niezawodność – kiedy blok ulega uszkodzeniu, tracony jest wskaźnik do reszty pliku

Przydział indeksowy

- plik ma własny blok indeksowy, który jest tablicą adresów bloków dyskowych przydzielonych danemu plikowi



Zalety:

- uniknięcie wad przydziału ciągłego
- umożliwia dostęp bezpośredni do pliku

Wady:

- marnotrawstwo przestrzeni
- na wskaźniki w bloku indeksowym należy poświęcić więcej miejsca niż na wskaźniki w przydziale listowym
- kwestia rozmiaru bloku indeksowego