

Wykład 7

Zarządzanie pamięcią

A. Podstawy

1. Wiązanie adresów

- Czas kompilacji
- Czas ładowania
- Czas wykonania

2. Ładowanie dynamiczne

3. Łączenie dynamiczne

B. Wymiana

C. Przydział pojedynczego obszaru

D. Przydzielenie wielu obszarów (pierwsza pasująca, najlepiej, najgorzej pasująca)

E. Stronicowanie (sprzęt, planowanie, długoterminowe, implementacja, ochrona)

A.

Pamięć jest wielką tablicą poadresowanych słów (lub bajtów). Współpraca z pamięcią polega na tworzeniu ciągu operacji czytania lub pisania odnoszących się do konkretnych adresów pamięci.

Np. typowy cykl wykonania rozkazu rozpoczyna się od pobrania go z pamięci. Rozkaz jest następnie dekodowany i może wymagać pobrania z pamięci argumentów. Po wykonaniu rozkazu jego wyniki mogą zostać zapisane do pamięci.

UWAGA: Do pamięci dociera tylko strumień kolejnych adresów pamięci oraz zlecenie rodzaju operacji (zapis, odczyt). Pamięć nie otrzymuje żadnych informacji o tym jak te adresy są wytwarzane (licznik rozkazów, adresowanie pośrednie, indeksowanie).

1. Wiązanie adresów

Przed wykonaniem procesu trzeba go wprowadzić do pamięci operacyjnej. Proces użytkownika może rezydować w dowolnej części pamięci fizycznej. Program użytkownika zanim zostanie wykonany przechodzi przez kilka faz.(niektóre mogą nie zawsze występować) .Podczas tych faz reprezentacja adresów może ulegać zmianie.

- Jeśli podczas kompilacji znane jest miejsce, w którym proces będzie przebywał w pamięci, to generować można kod bezwzględny (w MS DOS pliki *.com zawierają programy z adresami bezwzględnymi)
- Kompilator może też wytwarzać kod przemieszczalny (relokowalny).Określany jest w czasie ładowania adres startowy. Inne adresy są względem niego obliczane w czasie ładowania
- Jeżeli proces może ulegać przemieszczeniom w pamięci(podczas wykonania) to wiązanie adresów może być wykonane dopiero w czasie wykonywania (wymaga to specjalnego wyposażenia sprzętowego)

2. Ładowani dynamiczne

Podprogram nie jest ładowany do pamięci dopóki nie jest wywoływany. Wszystkie podprogramy są w postaci przemieszczalnej przemieszczane na dysku

3. Łączenie dynamiczne

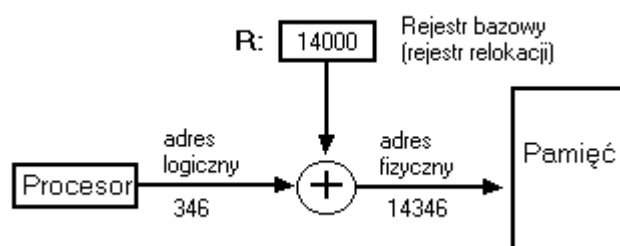
Tu opóźnia się łączenie. Cecha ta dotyczy zazwyczaj bibliotek systemowych(np. bibliotek języków programowania) Ładuje się do pamięci- ale łączy się później.

Przełączanie kontekstu (przykład obliczenia czasu)

Niech proces ma rozmiar 1 MB, pamięć pomocnicza=HDD o transferze 20 MB/s. Przesłanie 1MB zajmie 0,05s=50ms. Jeżeli średni czas oczekiwania=8ms to czas wymiany zajmie 58ms. Ponieważ należy przesłać w obie strony mamy 116ms. Aby wykorzystanie procesora mogło być dobre należy zadbać aby czas wykonania procesu był długi w stosunku do czasu wymiany. Np. przy planowaniu algorytmem rotacyjnym kwant czasu powinien być znacznie większy niż 5ms.

C) Przydział pojedynczego obszaru

- Najprostszym schematem zarządzania pamięcią jest brak schematu.
Zalety: użytkownik ma maksimum elastyczności
Wady: Brak jakichkolwiek usług. (kontroli przerw, wykonywania funkcji systemowych, obsługi błędów, wieloprogramowości)
- Podział pamięci na dwa rejony (dla użytkownika, dla systemu rezydującego)
Tu stosuje się ochronę sprzętową systemu operacyjnego (rej. bazowy; graniczny)
- Ładowanie procesów użytkownika do pamięci:
 - Jeżeli adres bazowy jest znany podczas kompilacji, to można generować kod bezwzględny.
 - Inne podejście, to opóźnienie wiązania do czasu wykonania. Wymaga to wsparcia sprzętowego:



Program użytkownika nie ma do czynienia z adresami fizycznymi

Program użytkownika działa na adresach logicznych

Ten sam sprzęt realizuje odwzorowanie adresów logicznych na fizyczne

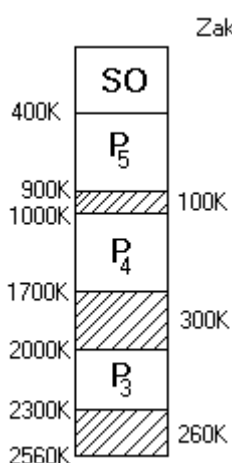
Adresy logiczne: $0 \div \text{max}$

Adresy fizyczne: $(R+0) \div (R+\text{max})$

Koncepcja logicznej przestrzeni adresów powiązanej z oddzielną od niej fizyczną przestrzenią adresów jest podstawą właściwego zarządzania pamięcią.

D) Przydzielanie wielu obszarów (sposoby)

W systemie wieloprogramowym w pamięci operacyjnej przebywa wiele procesów.



- Najprostszy sposób – podział pamięci operacyjnej na obszary o stałym rozmiarze. Każdy obszar zawiera jeden proces.
- Schemat podstawowy
- Pierwsza pasująca (wystarczającej wielkości)
- Najlepiej pasująca (najmniejsza z dostatecznie dużych dziur - lista dziur może być uporządkowana)

- Najgorzej pasująca – największa dziura (lista może być uporządkowana)

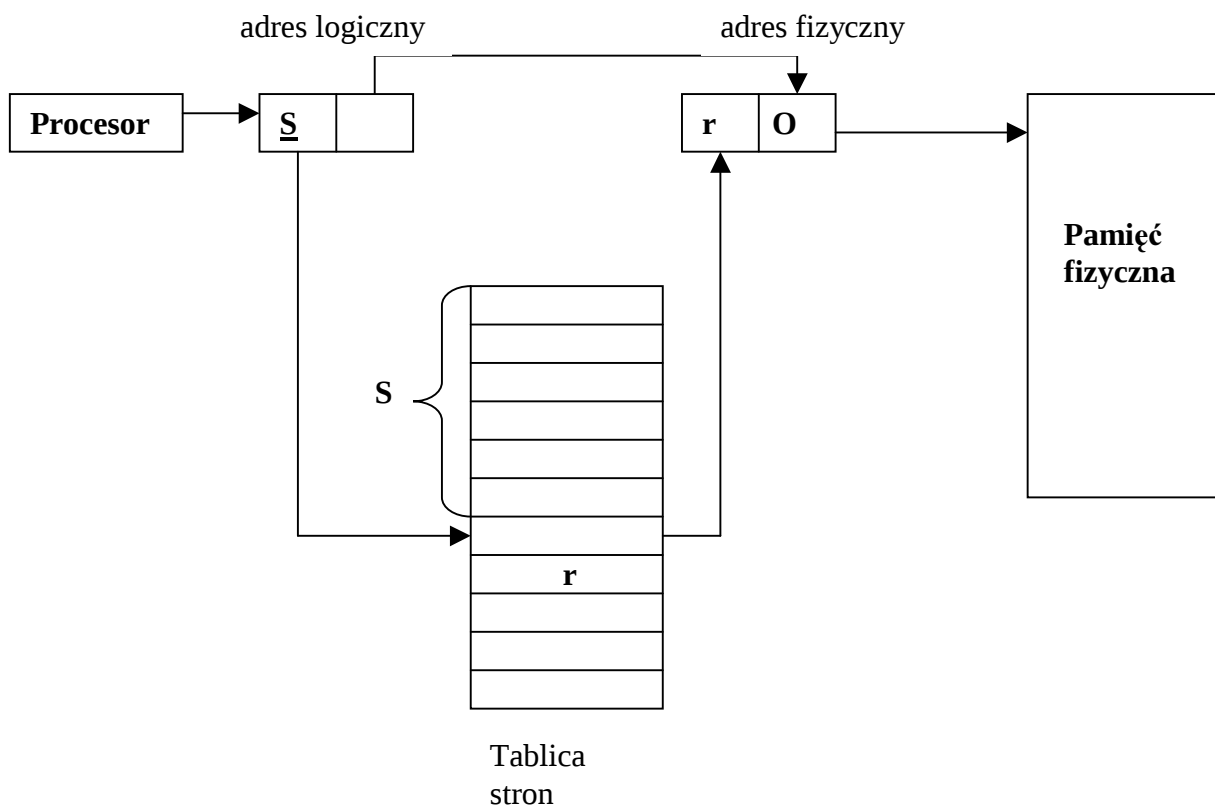
Zewnętrzna fragmentacja – dużo dziur o bardzo niewielkich rozmiarach.

Wewnętrzna fragmentacja – dziura na ogół ma inny rozmiar niż potrzebny dla procesu.

E) Stronicowanie

Stronicowanie jest metodą pozwalającą na to by pamięć przydzielona procesowi była nieciągła.

Pamięć fizyczną dzieli się na ramki o stałej długości (256 B – 4 kB →). Pamięć logiczna dzieli się na bloki takiego samego rozmiaru zwane stronami. Gdy ma nastąpić wykonanie procesu, wówczas jego strony przebywające w pamięci pomocniczej, są wprowadzane w dowolne (wolne) ramki pamięci fizycznej. Pamięć pomocniczą dzieli się na bloki tego samego rozmiaru co ramki w pamięci operacyjnej. Sprzętowe zaplecze stronicowania pokazano na rysunku:



(S,O) – adres wygenerowany przez procesor

S – numer strony

O – odległość na stronie

S – używane jako indeks w tablicy stron

Tablica stron – zawiera adresy bazowe wszystkich stron w pamięci operacyjnej

Uwaga:

Stronicowanie jest w istocie odmianą dynamicznego przemieszczania. Każdy adres logiczny zostaje powiązany za pośrednictwem sprzętowych środków stronicowania z jakimś adresem fizycznym.

Stronicowanie przypomina używanie tablicy rejestrów bazowych – po jednym na każdą ramkę pamięci.

Korzystanie ze stronicowania eliminuje fragmentację zewnętrzną. Pozostaje jednak wewnętrzna (średnio pół strony na proces).

Każdy system operacyjny ma własne metody przechowywania tablic stron. Większość systemów operacyjnych przydziela tablicę stron do każdego procesu (utrzymując w bloku kontrolnym procesu – wskaźnik do tej tablicy).

Tablice stron ze względu na ich wielkość przechowuje się w pamięci operacyjnej utrzymując w wraz z każdym obsługiwanym procesem wartość wskaźnika do jej początku w specjalnym rejestrze (rejestr bazowy tablicy stron).

Przełączanie kontekstu zmienia ten wskaźnik - proces otrzymuje wskaźnik do „swojej” tablicy stron (z bloku kontrolnego procesu)

Schemat taki wymaga dwóch kontaktów z pamięcią w celu uzyskania dostępu do jednego słowa (jeden - do pozycji w tablicy stron, drugi – do danego słowa $\Rightarrow$ patrz rysunek).

Powoduje to dwukrotne zwolnienie dostępu do pamięci. Rozwiązanie: rejestry asocjacyjne – mała pamięć o wyjątkowo dużej szybkości realizowana jako tablica kluczowana.

Sposób użycia rejestrów asocjacyjnych:

Procesor $\rightarrow$ adres logiczny $\rightarrow$ nr strony $S \rightarrow$ rejestr asocjacyjny (zawiera numery stron i odpowiadające im numery ramek) $\rightarrow$

$\rightarrow$ a) Jeśli jest $\rightarrow$ nr ramki $\rightarrow$ dostęp do pamięci fizycznej

$\rightarrow$ b) Jeśli nie $\rightarrow$ tablica stron $\rightarrow$ dodaj do rejestru asocjacyjnego $\rightarrow$ dostęp do pamięci

W przypadku a) czas zadania zwiększa się o 10 % a w b) o 100%

Procent numerów stron znajdujących w rejestrach asocjacyjnych - współczynnik trafień (bit ratio) ma wpływ na efektywny czas dostępu

$$\text{Efektywny czas dostępu} = HR * (MA + RAA) + (1 - HR) * (2 * MA + RAA),$$

gdzie :

HR – współczynnik trafień

MA – czas dostępu do pamięci

RAA – czas dostępu do rejestrów asocjacyjnych