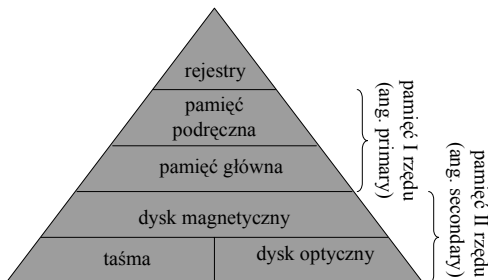


Zarządzanie pamięcią operacyjną — zagadnienia podstawowe

Pamięć jako zasób systemu komputerowego

- ☼ Pamięć jest zasobem służący do przechowywania danych.
- ☼ Z punktu widzenia systemu pamięć jest zasobem o strukturze hierarchicznej (począwszy od rejestrów procesora, przez pamięć podręczną, pamięć główną, skończywszy na pamięci masowej), w której na wyższym poziomie przechowywane są dane, stanowiące fragment zawartości poziomu niższego.
- ☼ Z punktu widzenia procesu pamięć jest zbiorem bajtów tablicą bajtów identyfikowanych przez adresy, czyli tablicą bajtów, w której adresy są indeksami.

Hierarchia pamięci



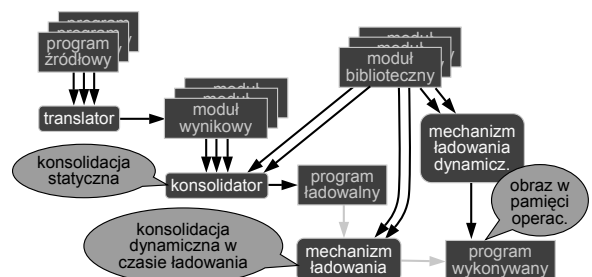
Przestrzeń adresowa

- ☼ Przestrzeń adresowa jest zbiór wszystkich dopuszczalnych adresów w pamięci.
- ☼ W zależności od charakteru adresu odróżnia się:
 - przestrzeń fizyczną — zbiór adresów przekazywanych do układów pamięci głównej (fizycznej).
 - przestrzeń logiczną — zbiór adresów generowanych przez procesor w kontekście aktualnie wykonywanego procesu.

Obraz procesu w pamięci

- ☼ Tworzenie obrazu
 - Kompilacja
 - Konsolidacja
 - Ładowanie kodu
- ☼ Relokacja
- ☼ Ochrona
- ☼ Współdzielenie

Tworzenie obrazu procesu



Wiązanie i przekształcanie adresów

- W modelu von Neumana adresy dotyczą rozkazów (instrukcji) oraz zmiennych (danych).
- Jeśli w programie źródłowym występują adresy, to mają one postać symboliczną (etykiety w assemblerze) lub abstrakcyjną (wskaźniki w C lub Pascalu).
- Adresy związane z lokalizacją pojawiają się na etapie translacji i są odpowiednio przekształcane aż do uzyskania adresów fizycznych.

Translacja

- W wyniku translacji (kompilacja, asemblacja) powstaje przemieszczalny moduł wynikowy (relocatable object module), w którym wszystkie adresy liczone są względem adresu początku modułu.
- Gdyby program składał się z jednego modułu, a jego docelowa lokalizacja w pamięci była z góry znana, na etapie translacji mogłyby zostać wygenerowane adresy fizyczne.

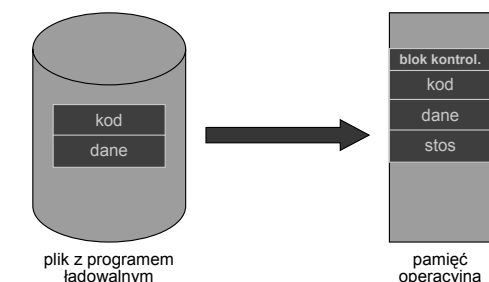
Konsolidacja

- Konsolidacja statyczna — odniesienia do innych modułów zamieniane są na adresy w czasie tworzenia programu wykonywalnego.
- Konsolidacja dynamiczna
 - w czasie ładowania — w czasie ładowania programu następuje załadowanie modułów bibliotecznych i wiązanie odpowiednich adresów,
 - w czasie wykonania — załadowanie modułów bibliotecznych i wiązanie adresów następuje dopiero przy odwołaniu się do nich w czasie wykonywania programu.

Konsolidacja statyczna

- W czasie łączenia modułów przemieszczalnych w jeden program ładowalny zwany też modulem absolutnym (ang. absolute module), do adresów przemieszczalnych dodawane są wartości, wynikające z przesunięcia danego modułu przemieszczalnego względem początku modułu absolutnego.
- Gdyby docelowa lokalizacja programu w pamięci była z góry znana, na etapie tym mogłyby zostać wygenerowane adresy fizyczne.

Ładowanie kodu (1)



Ładowanie kodu (2)

- Ładowanie absolutne — program ładowany jest w ustalone miejsce w pamięci, znane w momencie tworzenia programu ładowalnego.
- Ładowanie relokowalne — fizyczna lokalizacja procesu ustalana przy ładowaniu do pamięci.
- Ładowanie dynamiczne w czasie wykonania — fizyczna lokalizacja procesu w pamięci może ulec zmianie w czasie wykonywania.

Problem realizacji relokacji

- Jeśli adresy fizyczne ustalane są dopiero na etapie tworzenia obrazu procesu w pamięci, **wszystkie** adresy muszą zostać odpowiednio przeliczone przy ładowaniu programu do pamięci.
- Wymiana procesów pomiędzy pamięcią główną a pamięcią pomocniczą wymaga przeliczenia adresów przed każdym ponownym ładowaniem obrazu procesu do pamięci, gdyż trudno wymagać, żeby proces ładowany był dokładnie w ten sam obszar, który zajmował wcześniej.

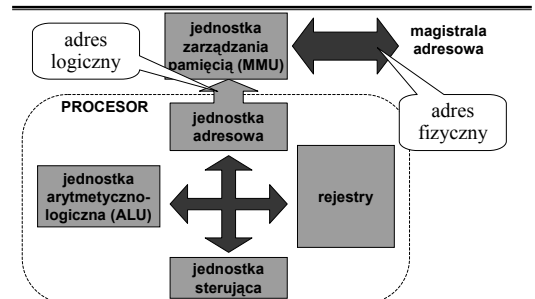
Wiązanie adresu w czasie ładowania

- Ustalanie adresów fizycznych w czasie ładowania wymaga wskazania tych miejsc w programie ładowalnym, które zawierają adresy absolutne i dodania do nich przemieszczenia względem początku obszaru pamięci fizycznej.
- Format pliku z programem ładowalnym musi zatem dodatkowo uwzględniać identyfikację adresów, przez co staje się skomplikowany i powoduje wzrost rozmiaru samego pliku.
- Przeliczanie adresów jest operacją czasochłonną.

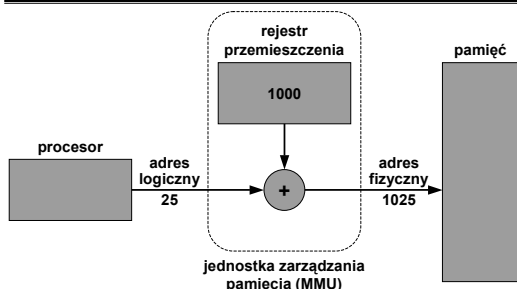
Wiązanie adresu w czasie wykonania

- Ustalanie adresów fizycznych w czasie wykonania oznacza, że adres wygenerowany przez procesora poddawany jest odpowiedniej transformacji, zanim zostanie wystawiony na szynie adresowej magistrali systemowej systemu.
- Efektywne generowanie adresów fizycznych w czasie wykonania wymaga odpowiedniego wspomagania sprzętowego i wprowadza podział na adresy logiczne i fizyczne.

Adres logiczny i fizyczny



Przykład odwzorowania adresu logicznego na fizyczny



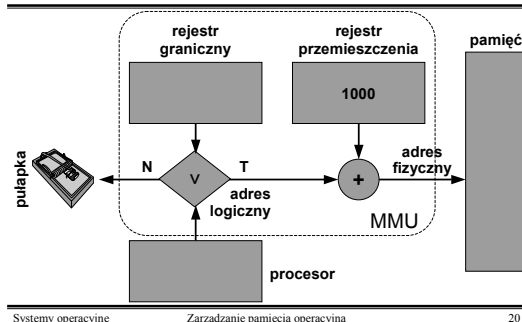
Ochrona pamięci

- W systemie wielozadaniowym występuje konieczność ochrony przed zamierzoną lub przypadkową ingerencją procesu w obszar innego procesu lub jądra systemu operacyjnego.
- Ochrona jądra systemu operacyjnego wskazana jest również w przypadku systemów jednozadaniowych, nie jest jednak elementem krytycznym, gdyż całość zasobów systemu przeznaczona jest na potrzeby jednego przetwarzania. Brak ochrony spowodować może jednak utratę kontroli nad systemem przez użytkownika.

Problemy realizacji ochrony pamięci

- Ochrona pamięci wymaga weryfikacji adresów generowanych przez proces przy każdorazowym odniesieniu do pamięci.
- W celu weryfikacji adresów w kontekście danego procesu muszą być przechowywane informacje na temat dostępności obszarów pamięci (zakres adresów, tryb dostępu).
- Efektywna realizacja ochrony wymaga odpowiedniego wspomaganie sprzętowego.

Przykład ochrony obszaru pamięci



Współdzielenie pamięci

- Efektywność wykorzystania pamięci
 - współdzielenie kodu programu
 - współdzielenie kodu funkcji bibliotecznych
- Kooperacja procesów
 - synchronizacja działań procesów
 - komunikacja pomiędzy procesami (współdzielenie danych)

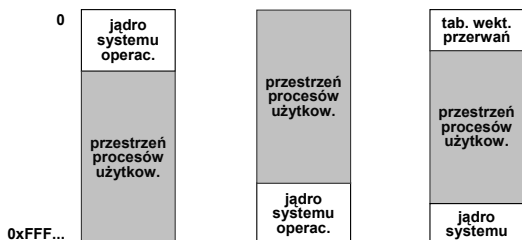
Problemy realizacji współdzielenia

- Realizacja współdzielenia przy braku rozdziału pomiędzy fizyczną i logiczną przestrzenią adresową wymaga rozwiązywania podobnych problemów z wiązaniem adresów jakie pojawiają się przy relokacji, a ponadto uniemożliwia ochronę pamięci lub wprowadza ograniczenia w dostępie.
- Współdzielenie pamięci przy zachowaniu elastyczności dostępu wymaga rozdzielania logicznej i fizycznej przestrzeni adresowej.

Lokalizacja procesu w pamięci

- podział stały pamięci,
- podział dynamiczny pamięci,
- proste stronicowanie,
- prosta segmentacja,
- stronicowanie w systemie pamięci wirtualnej,
- segmentacja w systemie pamięci wirtualnej.

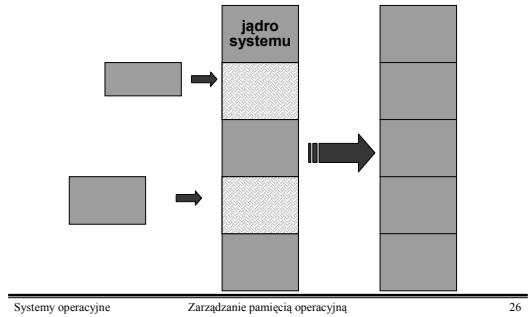
Ogólny obraz pamięci fizycznej



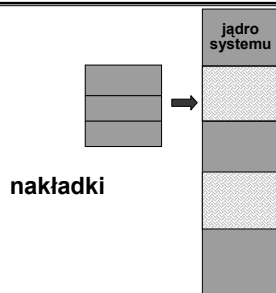
Podział stały

- Podział pamięci na stałe obszary (strefy, partycje), których rozmiar i położenie ustalane są na etapie konfiguracji systemu.
- Procesowi przydzielany jest cały obszar o rozmiarze większym lub równym rozmiarowi procesu.
- Zalety: łatwość implementacji i zarządzania
- Wady: słaba efektywność wykorzystania pamięci (fragmentacja wewnętrzna, ograniczona liczba aktywnych procesów).

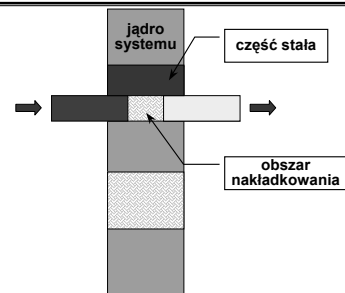
Podział stały — partycje o równym rozmiarze



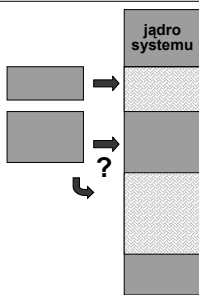
Podział stały — problem zbyt małych partycji



Nakładkowanie



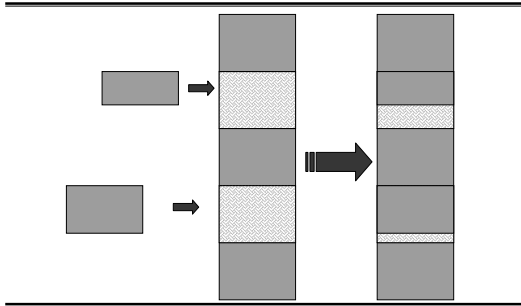
Podział stały — partycje o różnych rozmiarach



Podział dynamiczny

- Podział pamięci tworzony jest w czasie pracy systemu stosownie do żądań procesów.
- Proces ładowany jest w obszar o rozmiarze dokładnie odpowiadającym jego wymaganiom.
- Zalety: lepsze wykorzystanie pamięci (brak fragmentacji wewnętrznej)
- Wady: skomplikowane zarządzanie, wynikające z konieczności utrzymywania odpowiednich struktur danych w celu identyfikacji obszarów zajętych oraz wolnych.

Obraz pamięci przy podziale dynamicznym



Systemy operacyjne

Zarządzanie pamięcią operacyjną

31

Podział dynamiczny — problem wyboru bloku

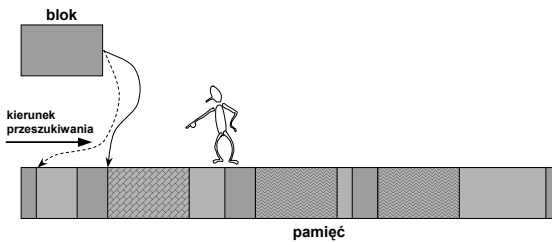
- ❖ Pierwsze dopasowanie — przydziela się **pierwszy wolny** obszar (tzw. dziurę) o wystarczającej wielkości. Poszukiwanie kończy się po znalezieniu takiego obszaru.
- ❖ Najlepsze dopasowanie — przydziela się **najmniejszy dostatecznie duży** wolny obszar pamięci. Konieczne jest przeszukiwanie wszystkich dziur.
- ❖ Następne dopasowanie — podobnie jak pierwsze dopasowanie, ale poszukiwania rozpoczyna się do miejsca ostatniego przydziału.
- ❖ Najgorsze dopasowanie — przydziela się **największy** wolny obszar pamięci. Konieczne jest przeszukiwanie wszystkich dziur.

Systemy operacyjne

Zarządzanie pamięcią operacyjną

32

Pierwsze dopasowanie (ang. first fit)

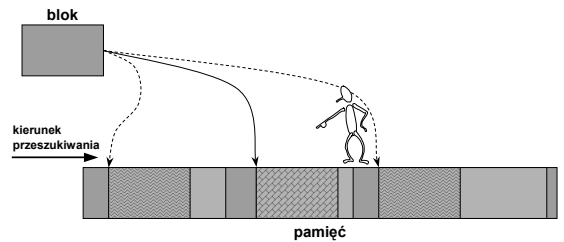


Systemy operacyjne

Zarządzanie pamięcią operacyjną

33

Najlepsze dopasowanie (ang. best fit)

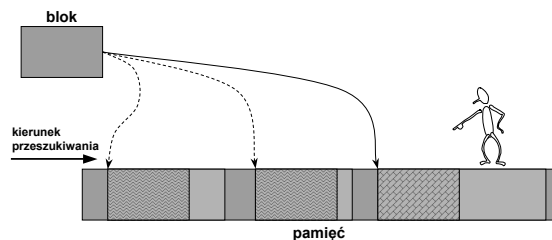


Systemy operacyjne

Zarządzanie pamięcią operacyjną

34

Najgorsze dopasowanie (ang. worst fit)



Systemy operacyjne

Zarządzanie pamięcią operacyjną

35

System bloków bliźniaczych (*buddy*)

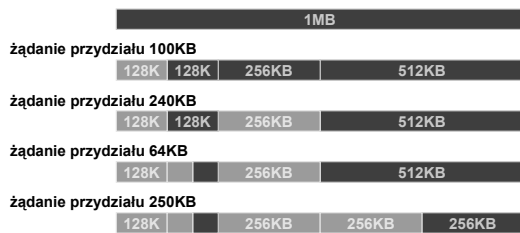
- ❖ Pamięć dostępna dla procesów użytkownika ma rozmiar 2^U .
- ❖ Przydzielany blok ma rozmiar 2^K , gdzie $L \leq K \leq U$.
- ❖ Początkowo dostępny jest jeden blok o rozmiarze 2^U .
- ❖ Realizacja przydziału obszaru o rozmiarze s polega na znalezieniu lub utworzeniu (przez połowienie) bloku o rozmiarze 2^i takim, że $2^{i-1} < s \leq 2^i$.

Systemy operacyjne

Zarządzanie pamięcią operacyjną

36

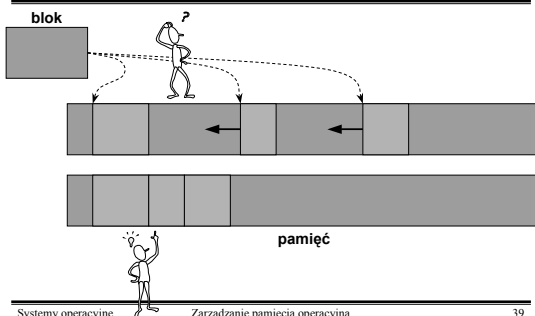
System *buddy* — przykład



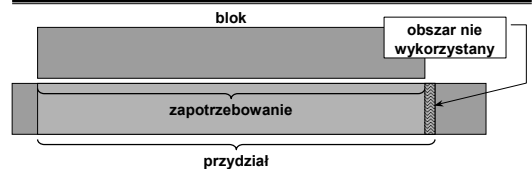
Fragmentacja

- Fragmentacja zewnętrzna — podział obszaru pamięci na rozłączne fragmenty, które nie stanowią ciągłości w przestrzeni adresowej (może to dotyczyć zarówno obszaru wolnego, jak i zajętego)
- Fragmentacja wewnętrzna — pozostawienie niewykorzystywanego fragmentu przestrzeni adresowej wewnątrz przydzielonego obszaru (formalnie fragment jest zajęty, w rzeczywistości nie jest wykorzystany)

Fragmentacja zewnętrzna



Fragmentacja wewnętrzna



Przydział dokładnie tylu bajtów, ile wynosi zapotrzebowanie, powoduje, że koszt utrzymania bardzo małego obszaru wolnego jest niewspółmiernie duży, np. dane o obszarze zajmują więcej bajtów, niż rozmiar tego obszaru. Dlatego wolny obszar przydzielany jest w całości, ale nie jest w pełni wykorzystany.

Proste stronicowanie (ang. paging)

- Arbitralny podział pamięć jest na ramki (ang. frames) w które ładowane są odpowiednie strony wynikające z podziału obrazu procesu.
- Podział logicznej przestrzeni adresowej na strony (ang. pages) o takim samym rozmiarze, jak ramki w pamięci fizycznej.
- Zalety:
 - brak fragmentacji zewnętrznej,
 - wspomaganie dla współdzielenia i ochrony pamięci.
- Wady: fragmentacja wewnętrzna (niewielka).

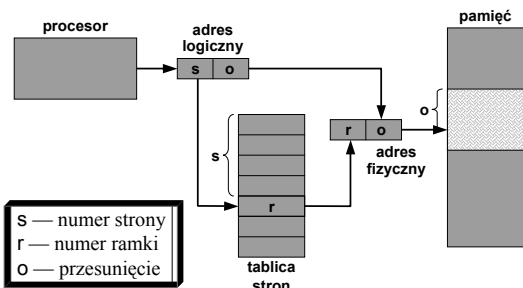
Stronicowanie — transformacja adresu

- Adres logiczny zawiera numer strony i przesunięcie na stronie (ang. offset).

nr strony (22 bity)	przesunięcie (10 bitów)
------------------------	----------------------------

- Transformacja adresu polega na zastąpieniu numeru strony numerem ramki.
- Odwzorowanie numeru strony na numer ramki wykonywane jest za pomocą tablicy stron (ang. page table).

Schemat transformacji adresu w systemie pamięci stronicowanej

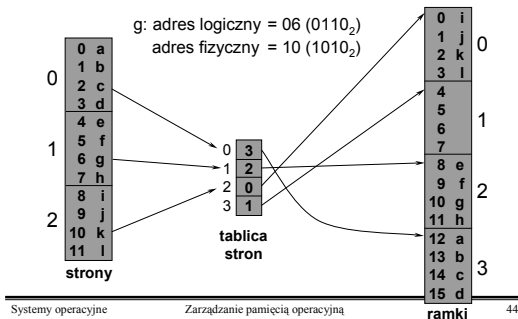


Systemy operacyjne

Zarządzanie pamięcią operacyjną

43

Przykład odwzorowania stron w ramki

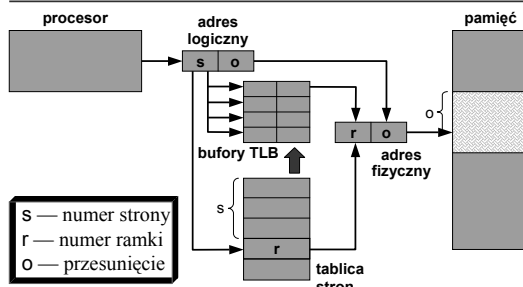


Systemy operacyjne

Zarządzanie pamięcią operacyjną

44

Bufory translacji adresów stron (TLB)



Systemy operacyjne

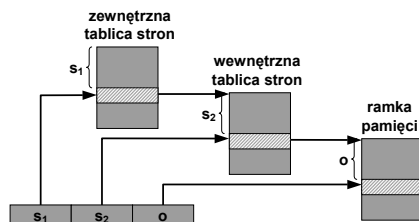
Zarządzanie pamięcią operacyjną

45

Stronicowanie wielopoziomowe

Budowa adresu

nr strony zewnętrznej	nr strony wewnętrznej	przesunięcie na stronie
-----------------------	-----------------------	-------------------------



Systemy operacyjne

Zarządzanie pamięcią operacyjną

46

Odwrócona tablica stron (ang. inverted page table)

- W odwróconej tablicy stron jest jedna pozycja dla każdej ramki.

PID procesu	nr strony	przesunięcie
-------------	-----------	--------------

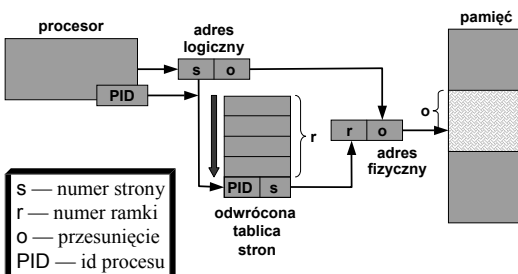
- Tablica stron indeksowana jest numerem ramki, a zawiera identyfikator procesu oraz wirtualny numer strony.

Systemy operacyjne

Zarządzanie pamięcią operacyjną

47

Odwrócona tablica stron — transformacja adresu



Systemy operacyjne

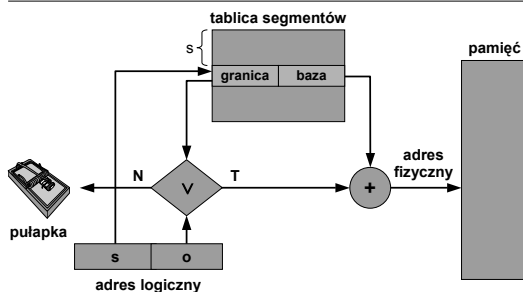
Zarządzanie pamięcią operacyjną

48

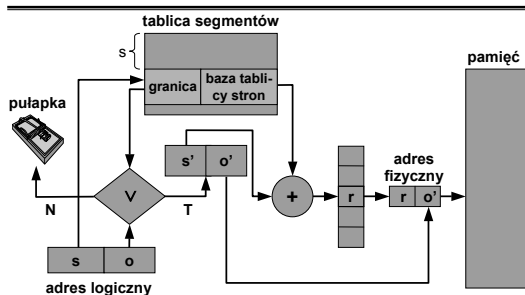
Segmentacja (ang. segmentation)

- Przestrzeń adresów logicznych jest postrzegana jako zbiór segmentów.
- Podstawowe atrybuty segmentu:
 - nazwa (identyfikator, numer), która jest indeksem w tablicy segmentów
 - adres bazowy — wskazanie na początek segmentu
 - rozmiar — długość segmentu w ustalonych jednostkach (np. w bajtach, paragrafach).
- Adres logiczny składa się z numeru segmentu i przesunięcia wewnątrz segmentu.
- Odwzorowanie adresów logicznych w fizyczne zapewnia tablica segmentów.

Schemat adresowania z segmentacją



Schemat adresowania z segmentacją i stronicowaniem



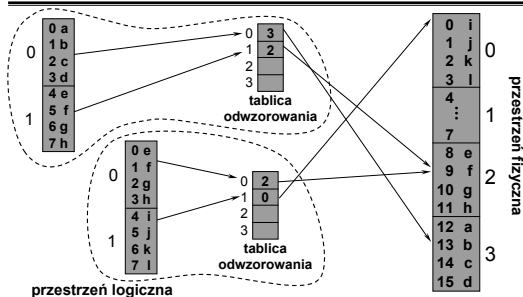
Prosta segmentacja

- Podział procesu na segmenty, które ładowane są do dynamicznych partycji w pamięci.
- Rozmieszczenie segmentów w pamięci może być dowolne (nie musi to być ciągły obszar)
- Zalety w porównaniu z podziałem dynamicznym: elastyczność w przydziale pamięci.
- Wady w porównaniu z podziałem dynamicznym: bardziej skomplikowane zarządzanie.

Odwzorowanie adresu logicznego na fizyczny — podsumowanie

- Przemieszczenie** — adres fizyczny powstaje z adresu logicznego przez dodanie przemieszczenia.
- Stronicowanie** — pamięć jest podzielona na logiczne strony o ustalonym rozmiarze odpowiadające fizycznym ramkom. Bardziej znacząca część bitów adresu interpretowana jest jako numer strony, a pozostała, mniej znacząca część bitów jest przesunięciem na stronie.
- Segmentacja** — pamięć jest podzielona na segmenty, czyli obszary zdefiniowane przez wskazanie początku w pamięci fizycznej i rozmiar. Adres logiczny obejmuje identyfikator (numer) segmentu oraz przesunięcie wewnątrz segmentu.

Współdzielenie pamięci przy rozdzielaniu fizycznej i logicznej przestrzeni adresowej



Ciągłość logicznej przestrzeni adresowej

- ☼ Czy w ogólnym przypadku logiczna przestrzeń adresowa w systemie stronicowania pamięci jest ciągła?
- ☼ Czy w ogólnym przypadku logiczna przestrzeń adresowa w systemie segmentacji pamięci jest ciągła?

