



Systemy Operacyjne

Pamięć główna

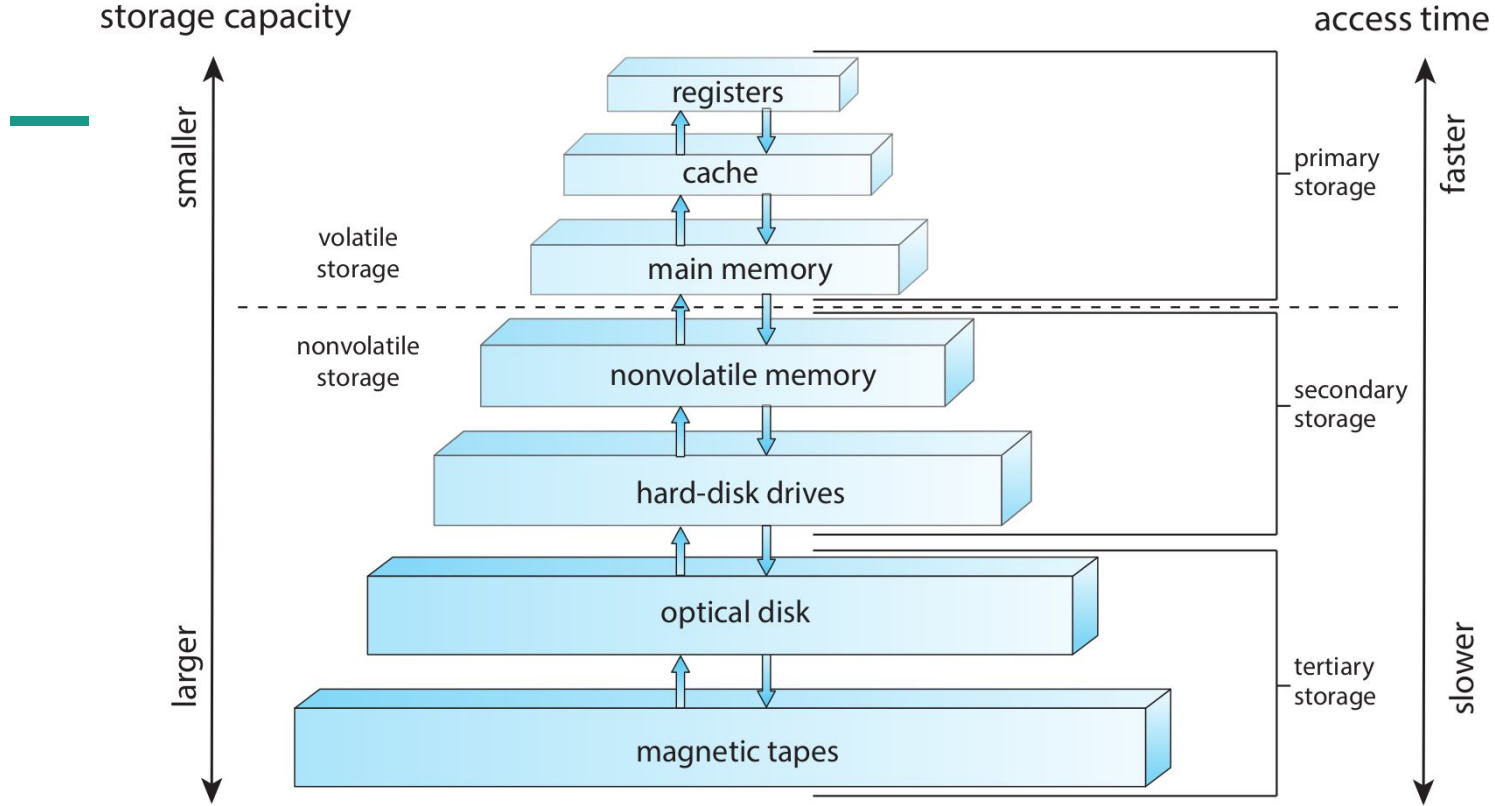
Dr hab. inż. Krzysztof Rzecki, prof. AGH

Na podstawie: Abraham Silberschatz, *Koncepcje systemów operacyjnych*



Współdzielenie zasobów

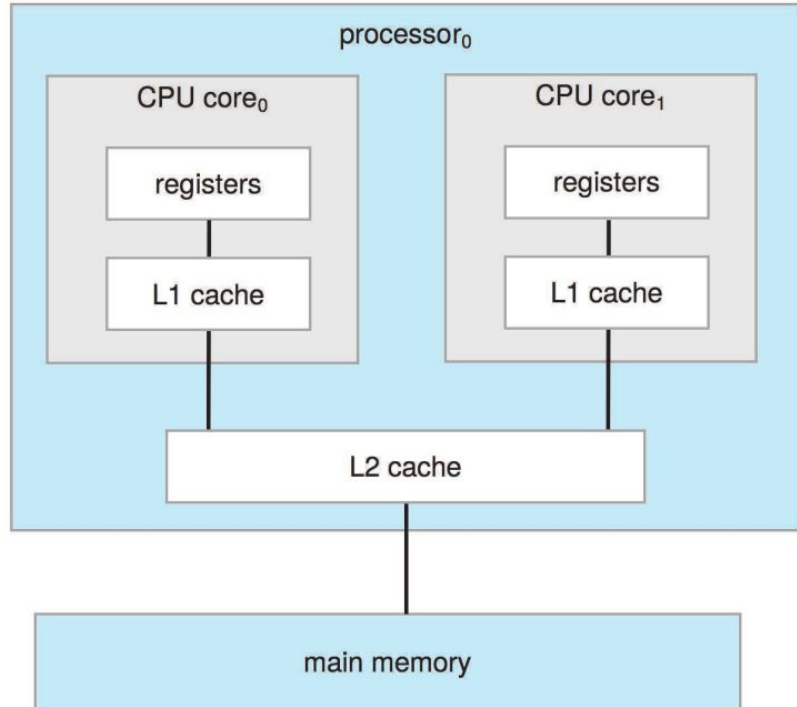
- CPU: współdzielenie w czasie
- MEM: współdzielenie w ilości
- HDD: współdzielenie w dostępie
- NET: współdzielenie w czasie



Źródło: A. Silberschatz, *Operating Systems Concepts Essentials*

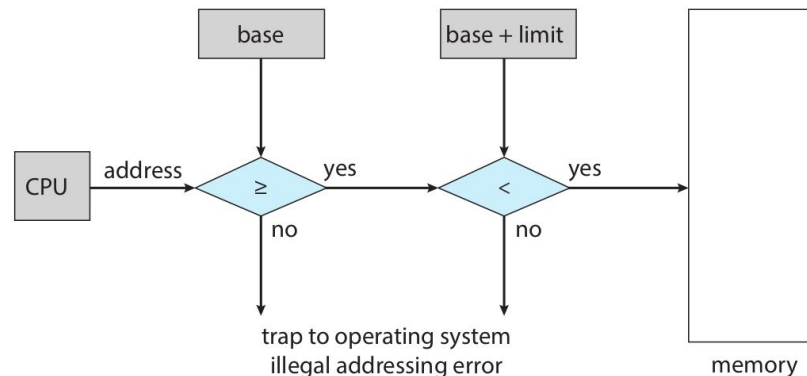
Procesor + pamięć

- Procesor ładuje instrukcje tylko z **pamięci głównej**, więc każdy program musi być do niej najpierw załadowany.
- Pamięć główna (ang. *main memory*, RAM - ang. *random-access memory*) wykonana jest w technologii półprzewodnikowej zwanej DRAM - ang. *dynamic random-access memory*.
- Nie wszystko mieści się w pamięci RAM oraz pamięć ta jest ulotna, stąd wymagana jest pamięć dodatkowa, tj. ang. *secondary storage* (ang. *hard-disk drives* - HDDs lub ang. *nonvolatile memory* - NVM).



Podstawy adresowania

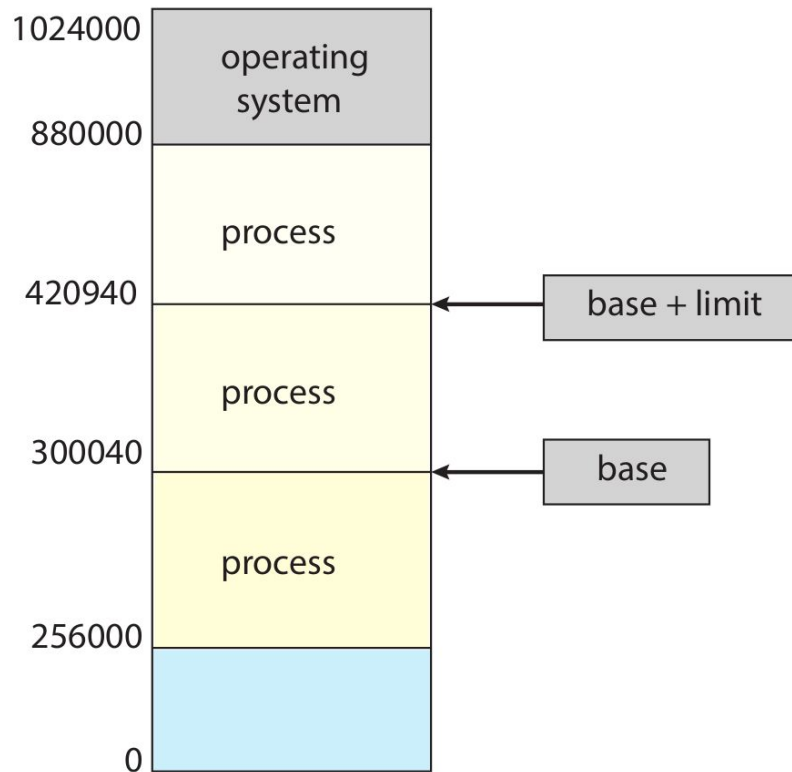
- Pamięć główna oraz rejestry wbudowane w CPU to jedyna pamięć, którą CPU może adresować bezpośrednio.
- Jeśli dana, na której mają być wykonywane operacje znajduje się na którejkolwiek z pozostałych pamięci, musi zostać najpierw skopiowana w obszar o bezpośrednim dostępie.
- Adresowanie rejestrów, w odróżnieniu od adresowania pamięci głównej, zazwyczaj odbywa się w jednym takcie procesora.
- Aby w trakcie uzyskiwania dostępu do pamięci głównej procesor nie marnował cykli, wykorzystywany jest cache.



Źródło: A. Silberschatz, *Operating Systems Concepts Essentials*

Izolacja pamięci

- Izolacja pamięci dla każdego procesu gwarantuje indywidualną przestrzeń.
- Pamięć przydzieloną procesowi wyznaczają dwa rejestry: baza i limit.
- Proces użytkownika (*user mode*) może zapisywać i odczytywać tylko pamięć w przydzielonym zakresie.
- Dlaczego ?
- System operacyjny (*kernel mode*) może swobodnie operować po całej pamięci, w szczególności zmieniać bazę i limit.
- Jak to ma zastosowanie ?



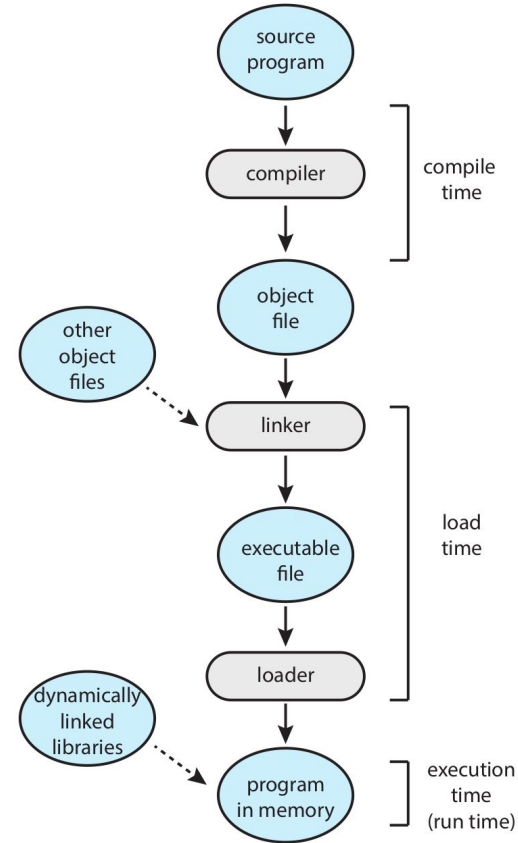
Źródło: A. Silberschatz, *Operating Systems Concepts Essentials*

Wiązanie adresu

- Adres w kodzie programu może być symboliczny, np. nazwa zmiennej.
- Kompilator wiąże w/w adres symboliczny do adresu relokowalnego (względem danego modułu).
- Linker lub loader zamienia adres relokowalny w bezwzględny.

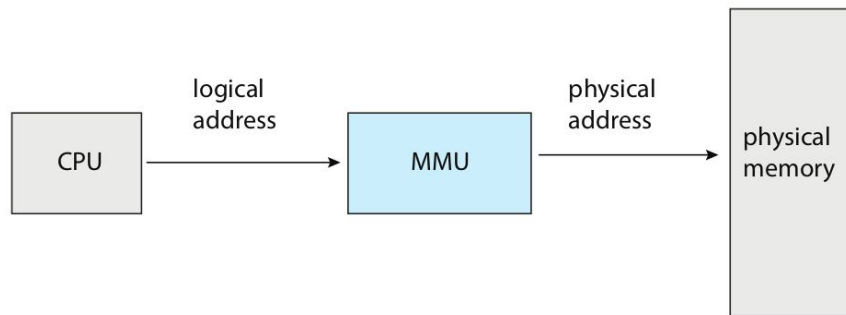
Wiązanie może wystąpić na każdym z etapów:

- Kompilacja (stare systemy),
- Ładowanie (MS-DOS),
- Wykonywanie - najczęściej (obecnie).



Logiczna i fizyczna przestrzeń adresowa

- Adres logiczny - adres widziany przez proces i dla niego dostępny.
- Adres fizyczny - adres na szynie adresowej.
- Translacją adresów zajmuje się MMU - ang. *Memory management unit*.
- Wiązanie adresów w czasie:
 - kompilacji -> adres logiczny == fizyczny,
 - ładowania -> adres logiczny == fizyczny,
 - wykonywania -> adres logiczny != fizyczny.
- Adres wirtualny = adres logiczny w sytuacji wiązania w czasie wykonywania.



Źródło: A. Silberschatz, *Operating Systems Concepts Essentials*

Ćw.: w C/C++ statyczna/dynamiczna rezerwacja pamięci, odczyt adresu zmiennej, kilkukrotne uruchomienie programu.



Dynamiczne ładowanie funkcjonalności

- Po stronie programisty leży konstrukcja programu z dynamicznie ładowanymi funkcjonalnościami.
- Funkcjonalności ładowane są dopiero w momencie ich wywołania.
- Kiedy któraś funkcjonalność wywołuje inną, a ta nie jest załadowana do pamięci, następuje jej załadowanie.
- Zaletą jest ładowanie funkcjonalności tylko wtedy, kiedy są potrzebne.



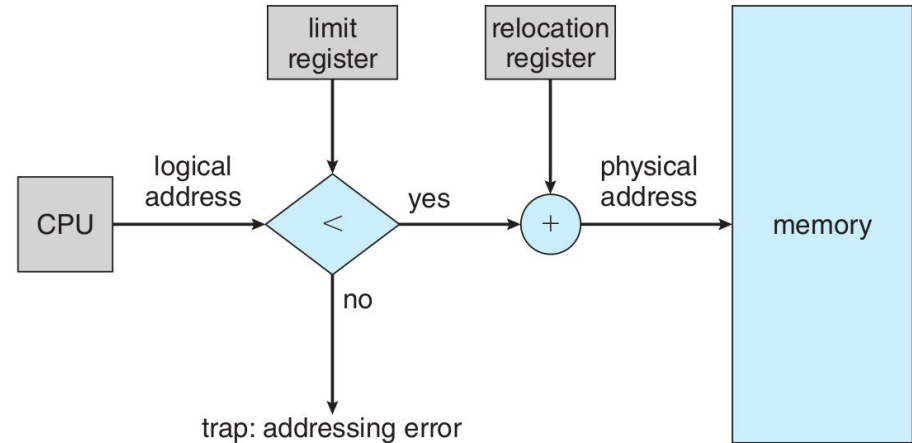
Linkowanie dynamiczne i statyczne

- DLL - ang. *Dynamic-Link Library* - biblioteka systemowa linkowana do programu w momencie jego uruchomienia (wymaga wsparcia ze strony systemu operacyjnego).
- Zalety DLL:
 - nie trzeba kopiować/implementować istniejących już modułów programu (oszczędność pamięci),
 - biblioteka może zostać jednokrotnie załadowana do pamięci, a wiele procesów może z niej korzystać,
 - biblioteki mogą być aktualizowane jednokrotnie, jakkolwiek każdy program może używać wskazanej wersji.
- Linkowanie statyczne - włączanie biblioteki systemowej do obrazu binarnego programu.

Pytanie C/C++: jak linkować statycznie/dynamicznie ?

Ochrona pamięci

- Relocation register (rejestr bazowy, rejestr relokacji) - początkowy adres fizyczny obszaru przeznaczanego dla procesu.
- Limit register - zawiera zakres adresów logicznych.
- Każdy proces wskazany przez scheduler CPU do uruchomienia sprawdzany jest względem w/w rejestrów.
- Dzięki temu można chronić system operacyjny i inne programy przed modyfikacją przez ten program.

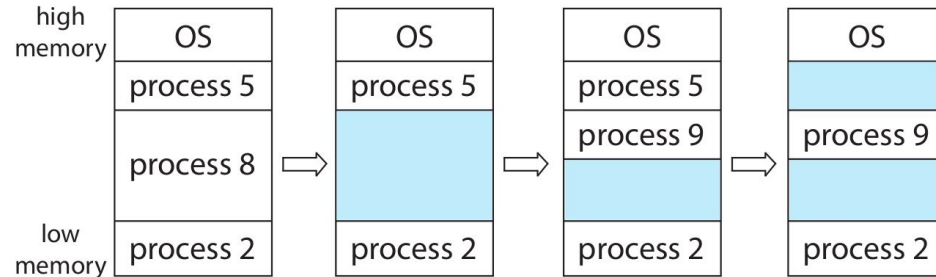


Źródło: A. Silberschatz, *Operating Systems Concepts Essentials*

Ćw.: odczyt wskaźnika spoza przydziel. pamięci.

Alokacja pamięci

- Metoda *variable-partition* - przypisanie procesu do zakresu pamięci i pamiętanie przez SO, które części pamięci są zajęte, a które wolne.
- Jeśli nie ma możliwości umieścić procesu w pamięci: a/ proces nie jest uruchamiany, b/ proces oczekuje w kolejce.
- Zarządzanie wolną przestrzenią (ang. *hole*) obejmuje jej dzielenie i łączenie i nazywane jest ang. *dynamic storage allocation problem*:
 - First fit - pierwszy pasujący
 - Best fit - najlepiej pasujący
 - Worst fit - największy wolny



Źródło: A. Silberschatz, *Operating Systems Concepts Essentials*

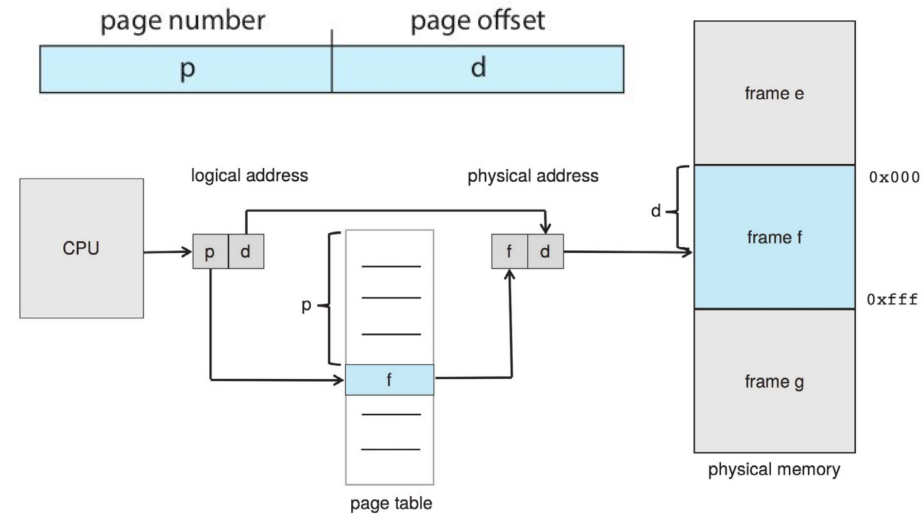


Fragmentacja

- W przypadku strategii first-fit i best-fit szybko dochodzi do tzw. fragmentacji zewnętrznej.
- Najgorszy przypadek fragmentacji: między każdą dwójką procesów jest wolna przestrzeń.
- Oprócz doboru strategii znaczenie ma też położenie nowego procesu (na początku czy na końcu wolnego miejsca?).
- Statystycznie, po pewnym czasie strategii first-fit aż 50% bloków zmarnowanych będzie na fragmentację.
- W przypadku podzielenia pamięci na bloki przydzielona pamięć dla danego procesu może być większa niż proces wymaga - różnica między wielkością przydzieloną a wymaganą to fragmentacja wewnętrzna.
- Rozwiązanie: kompaktowanie (Java: garbage collection) - możliwe tylko w czasie działania.
- Rozwiązanie: stronicowanie (następny slajd).

Stronicowanie, ang. *paging*

- Stronicowanie - logiczna przestrzeń adresowa jest nieciągła.
- Pamięć fizyczna dzielona jest na bloki, ang. *frames*.
- Pamięć logiczna dzielona jest na bloki tej samej wielkości, ang. *pages*.
- Kiedy proces ma być wykonywany, jego strony ładowane są do ramek dostępnej pamięci.
- Każdy proces ma swoją tablicę stron.
- Rozmiar strony zależny jest od sprzętu i wynosi od 4KB do 1GB i ze względu na adresację jest potęgą 2 (getconf PAGESIZE).

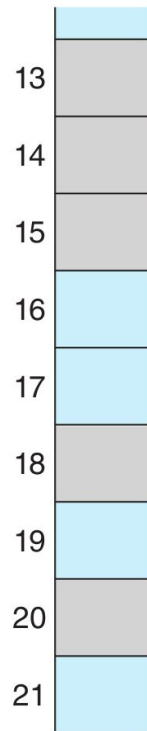
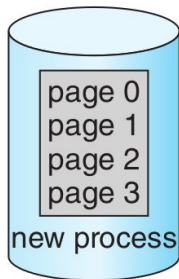


Źródło: A. Silberschatz, *Operating Systems Concepts Essentials*

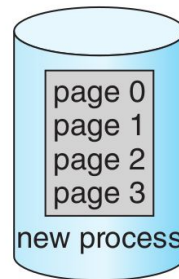
Alokacja pamięci

- Programista jest zupełnie odseparowany od widoku na pamięć fizyczną.
- Program może być rozrzucony po obszarach pamięci.
- Tablicą stronicowania oraz tablicą ramek zarządza system operacyjny.

free-frame list
14
13
18
20
15

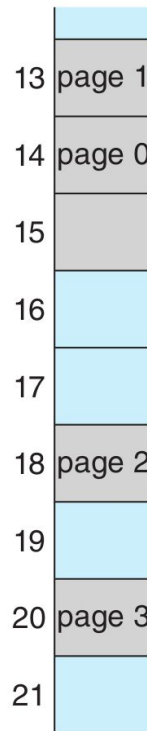


free-frame list
15



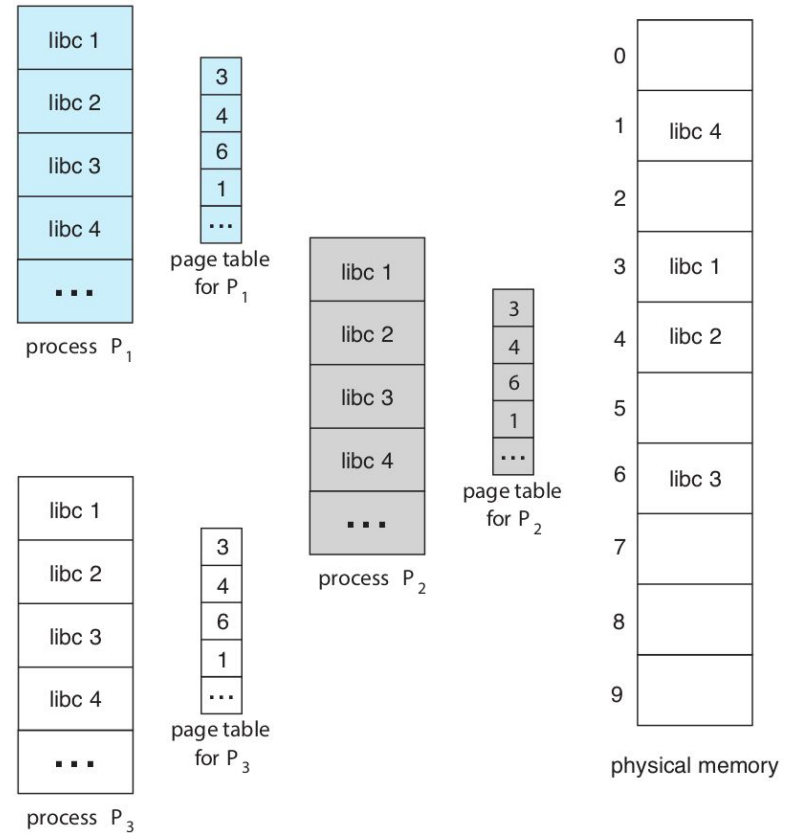
0	14
1	13
2	18
3	20

new-process page table

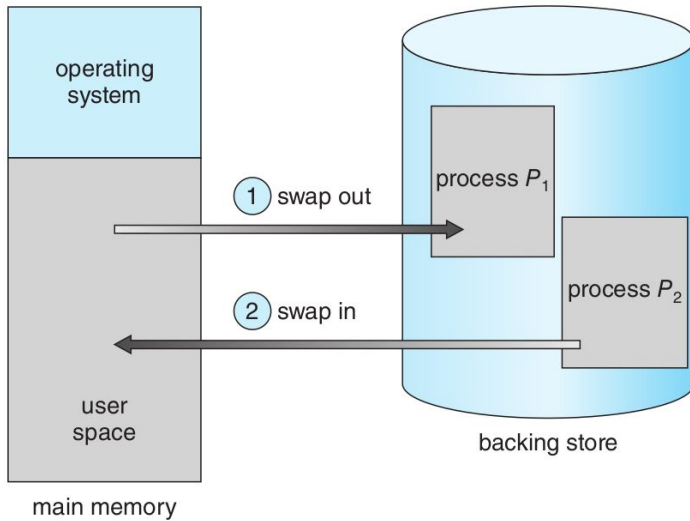


Współdzielenie

Przykład kodu wtórnego: biblioteka `libc`

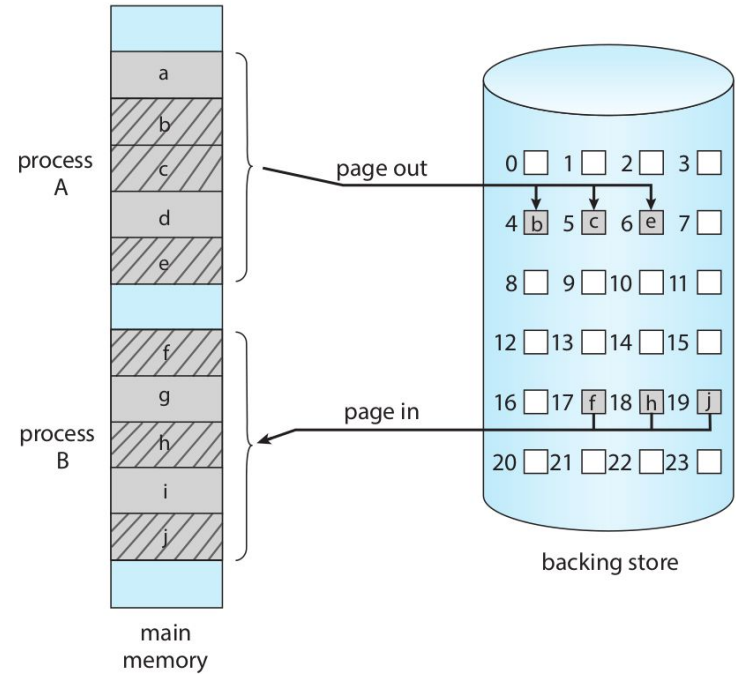


Swap



Źródło: A. Silberschatz, *Operating Systems Concepts Essentials*

Kiedyś



Źródło: A. Silberschatz, *Operating Systems Concepts Essentials*

Obecnie



Zadanie

Zakładając logiczną przestrzeń adresową składającą się z 64 stron o rozmiarze 1024 słów zmapowaną na fizyczną przestrzeń adresową o rozmiarze 32 ramek:

- Jaki rozmiar ma adres logiczny ?
- Jaki rozmiar ma adres fizyczny ?



Skrzyczne