

Systemy operacyjne Wykład 06

Wersja 2024

dr inż. Marek Wilkus <http://home.agh.edu.pl/~mwilkus>
Wydział Inżynierii Metali i Informatyki Przemysłowej
AGH Kraków

Na podstawie programu opracowanego przez dr inż. Krzysztofa Wilka

1

Zarządzanie pamięcią

2

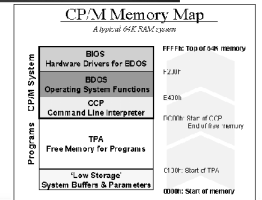
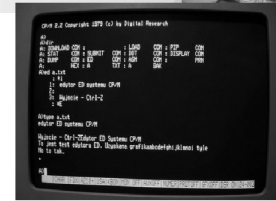
Zarządzanie pamięcią

- Przed wykonaniem program musi być pobrany z dysku i załadowany do pamięci.
- Tam działa jako proces.
- Podczas wykonywania, proces pobiera rozkazy i dane z pamięci.
- Większość systemów pozwala procesowi użytkownika przebywać w dowolnej części pamięci fizycznej.
- System musi uwzględniać to, że program „nie wie” pod jakim adresem pamięci będzie umieszczony. System musi zawierać mechanizmy „tłumaczące” adresy w programie na rzeczywiste adresy w pamięci fizycznej.
- System musi też gospodarować wolną pamięcią, racjonalnie ładując kolejne programy w wolne miejsca pamięci.

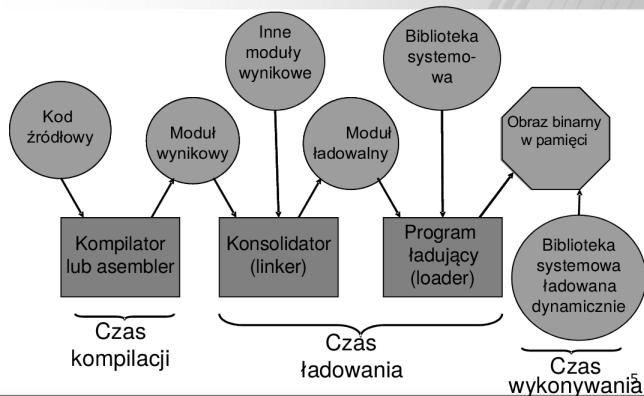
3

Rys historyczny

- 198x – CP/M – system z ładowaniem programów do pamięci.
 - TPA – wymienna strefa pamięci dla programów.
 - Ładowanie następnego programu powoduje nadpisanie pozostałości poprzedniego w TPA.
 - Iluzja wieloprogramowości – bankowanie regionów TPA.



Kod źródłowy → proces



Powiązanie programu z adresami w pamięci

- Może być dokonane w każdej z trzech faz:
 - Czas **kompilacji** - jeśli podczas kompilacji wiadomo pod jakim adresem pamięci program będzie przebywał, to tworzy się **kod bezwzględny**, np programy .com w DOS-ie, sterowniki w Windows. Aby zmienić położenie takiego programu w pamięci, trzeba go powtórnie przekompilować. Jeśli kod programu ma być **przemieszczalny**, to adresy muszą być określone w sposób względny, np w odległościach od początku modułu.
 - Czas **ładowania** - jeśli podczas kompilacji nie określono rzeczywistych adresów pamięci, to program ładujący wylicza je na podstawie adresów względnych.
 - Czas **wykonywania** - jeśli proces może ulegać przemieszczeniu w pamięci podczas wykonywania, to muszą istnieć mechanizmy wyliczania w dowolnym momencie rzeczywistych adresów.

6

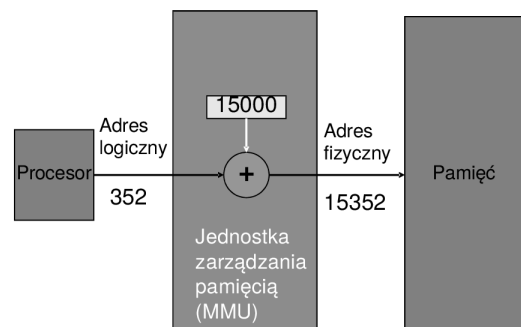
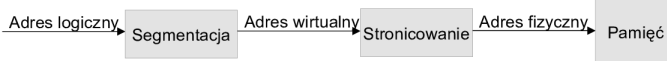
- **Ładowanie dynamiczne** - podprogram nie jest wprowadzany do pamięci dopóty, dopóki nie zostanie wywołany. Do pamięci wprowadza się jedynie program główny, a potem sukcesywnie potrzebne moduły. Dzięki temu oszczędza się miejsce w pamięci nie ładując nie potrzebnie wielkich modułów, np. obsługi błędów.
- **Konsolidacja dynamiczna** - Dynamiczne dołączanie bibliotek zewnętrznych - bez tej właściwości wszystkie programy muszą mieć dołączone kopie bibliotek, w tym systemowych. Powoduje to marnotrawstwo dysku i pamięci. Biblioteki dynamicznie linkowane są sprowadzane do pamięci w momencie ich wywołania i tam mogą służyć nawet kilku programom. Dodatkowa zaleta - aktualizacja biblioteki nie wymaga zazwyczaj przekompilowania programu.

- Potrzebne gdy proces jest większy niż ilość dostępnej pamięci.
- Przykład: Dwuprzebiegowy assembler:
Mamy do dyspozycji 150kB pamięci, elementy zadań mają rozmiary:
 - Kod przebiegu 1: 70kB
 - Kod przebiegu 2: 80kB
 - Tablice pomocnicze symboli: 20kB
 - Wspólne podprogramy: 30kB
- Razem: 200kB

- Załóżmy, że system nie umożliwi pamięci wirtualnej. Robimy dwie nakładki:
 - Tablica symboli, wspólne podprogramy, kod przebiegu 1, moduł obsługi nakładek (10kB): Razem 130kB.
 - Tablica symboli, wspólne podprogramy, kod przebiegu 2, moduł obsługi: Razem 140kB.
- Obydwie nakładki są poniżej 150kB.
 - Kody nakładek przechowywane są na dysku w postaci obrazów bezwzględnych pamięci i są ładowane przez moduł obsługi w zależności od potrzeb.
 - Program nakładkowy nie potrzebuje wsparcia ze strony systemu operacyjnego - wymaga starannego zaprogramowania programu obsługi.

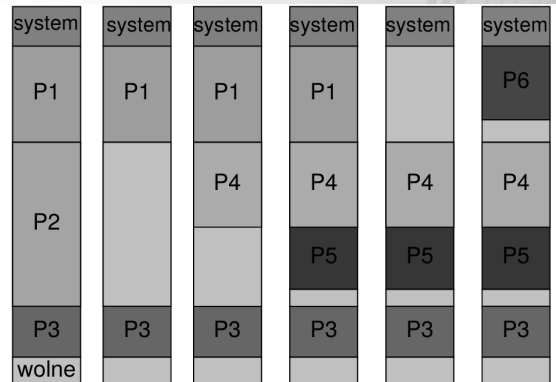
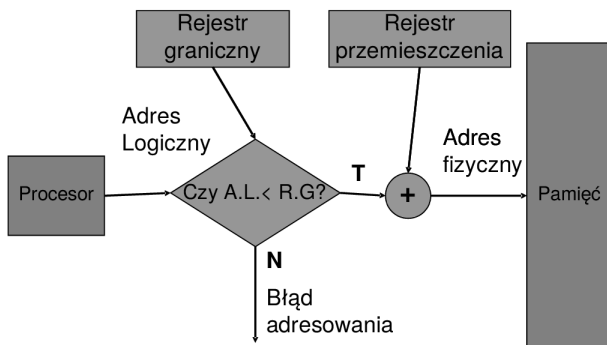
- **Adres fizyczny** - jest to adres oglądany przez jednostkę pamięci (umieszczony w jej rejestrze adresowym).
- **Adres logiczny** - jest to adres wygenerowany przez procesor.
- Odzworowanie adresów logicznych na fizyczne realizowane jest sprzętowo przez jednostkę zarządzania pamięcią (MMU).
- W MMU przeliczanie adresu odbywa się najczęściej poprzez dodanie do adresu z procesora wartości **rejestru przemieszczenia**.
- Program użytkownika działa wyłącznie na logicznych adresach.

- Pomiedzy adresem logicznym a fizycznym występuje jeszcze **adres wirtualny**.
- **Adres wirtualny** jest tym, czym intuicyjnie wydaje się być w starszych platformach adres fizyczny - czyli przesunięciem od początku przestrzeni adresowej.
- Ze względu na przełączanie banków pamięci, „rozdymanie” przestrzeni adresowej przełączając całe fragmenty pamięci, PAE, i dodatki w stylu LIM EMS, adres fizyczny może nie pokrywać się z adresem wirtualnym.



- Jest to tymczasowe odesłanie procesu do pamięci pomocniczej (na dysk) i przepisanie z powrotem w celu kontynuowania działania.
- Zazwyczaj po wymianie proces powraca na to samo miejsce w pamięci, chyba że są zapewnione mechanizmy przeliczania adresów.
- Warunkiem szybkiej wymiany jest używanie dysku o krótkim czasie dostępu i o pojemności zapewniającej pomieszczenie obrazów pamięci wszystkich użytkowników.
- W planowaniu priorytetowym stosuje się czasem wymianę poprzez **wytaczanie** procesu, gdy nadchodzi proces o wyższym priorytecie i **wtaczanie** z powrotem do pamięci, gdy procesy wysoko-priorytetowy skończył działanie.

- Pamięć operacyjna zajęta jest przez:
 - System operacyjny - umieszczony zazwyczaj w tej części pamięci, gdzie wektor przerwań - najczęściej pamięć "dolna".
 - Procesy użytkownika, umieszczane zazwyczaj w pamięci "górnej".
- W celu ochrony obszaru pamięci zajętego przez system, a także procesów użytkownika przed wzajemną ingerencją, wykorzystuje się **rejestr przemieszczenia i rejestr graniczny**:
 - Rejestr przemieszczenia - wartość najmniejszego adresu fizycznego dla danego procesu (offset),
 - Rejestr graniczny - maksymalny adres logiczny procesu.



Dziura - ciągły obszar niezajętej pamięci.

- **Pierwsze dopasowanie** - system przydziela pierwszą dziurę o wystarczającej wielkości. Szukanie rozpoczyna się od początku wykazu dziur lub od miejsca w którym zakończono ostatnie szukanie.
- **Najlepsze dopasowanie** - przydziela się najmniejszą z dostatecznie dużych dziur (najmniejsza pozostałość po przydziale).
- **Najgorsze dopasowanie** - przydziela się największą dziurę.
- Czasami duża pozostałość po takim przydziale jest bardziej przydatna niż małe fragmenty po najlepszym dopasowaniu.
- Ciekawe i oryginalne podejście do zagadnienia, ale praktyka wykazała, że dwie pozostałe metody są lepsze pod względem czasu działania i wykorzystania pamięci. 17

- **Fragmentacja zewnętrzna** - suma wolnych obszarów pamięci wystarcza na spełnienie zamówienia, ale nie tworzą one spójnego obszaru.
- **Fragmentacja wewnętrzna** - jeśli po przydzieleniu pamięci do procesu pozostałby wolny obszar wielkości kilku bajtów, to przydziela się go też do procesu, ale stanowi on „nieużytek” - nie będzie wykorzystany (ale zmniejszy się tablica „dziur”).
- Fragmentację zewnętrzną można zmniejszyć poprzez takie upakowanie procesów, aby cała wolna pamięć znalazła się w jednym dużym bloku. Jest to możliwe tylko wtedy, gdy ustalanie adresów jest wykonywane dynamicznie podczas działania procesu. Przetasaowanie procesów nie można robić podczas operacji we/wy.

Pomaga w racjonalnym wykorzystaniu wolnych miejsc w pamięci.

- Pamięć fizyczną dzieli się na bloki o stałej długości (ramki) o długości 2^n (np. 512 B do 16 MB).
- Pamięć logiczna podzielona jest na strony o tym samym rozmiarze.
- Wolną pamięć obrazuje lista wolnych ramek.
- Proces o wielkości N stron jest ładowany w N ramek (niekoniecznie kolejnych).
- Tablica stron odwzorowuje adresy logiczne na fizyczne.
- Eliminuje się fragmentację zewnętrzną, ale występuje fragmentacja wewnętrzna (zaokrąglenie w górę wielkości procesu do wielokrotności rozmiaru ramki).

Każdy adres wygenerowany przez procesor dzieli się na dwie części: numer strony i odległość na stronie. Numer jest używany jako indeks w tablicy stron, która zawiera adresy bazowe wszystkich stron w pamięci operacyjnej. Łącząc adres bazowy z odległością na stronie uzyskuje się fizyczny adres w pamięci.

Jeżeli rozmiar strony jest potęgą 2 oraz:

- rozmiar przestrzeni adresowej wynosi 2^m ,
- rozmiar strony wynosi 2^n ,

to m-n bardziej znaczących bitów adresu logicznego wskazuje nr strony ($=2^{(m-n)}$),

- n mniej znaczących bitów wskazuje odległość na stronie.

- Jest przechowywana w pamięci operacyjnej. Jej położenie wskazuje rejestr bazowy tablicy stron.
- Rozmiar tablicy stron jest przechowywany w rejestrze długości tablicy stron. Określa on na największy dopuszczalny adres.
- Przy korzystaniu z tablicy stron, dostęp do pamięci wymaga **dwukrotnego** dostępu do pamięci - dlaczego?
- W celu przyspieszenia dostępu do pamięci stosuje się rozwiązanie sprzętowe - małą, szybką pamięć podręczną zwaną **rejestrami asocjacyjnymi** lub **buforami translacji adresów stron**. Bufory te zawierają 8 do 2048 pozycji.
- Jeśli dany numer strony nie znajduje się w buforach, to przeszukiwana jest cała tablica stron. Przy dobrze skonstruowanym algorytmie, w buforach translacji znajduje się 80 do 98 % potrzebnych numerów stron.

- Jest to średni czas dostępu do każdego adresu pamięci.
- Przykład:
 - Przełknięcie rejestrów asocjacyjnych trwa 20ns
 - Dostęp do pamięci trwa 100 ns
- Współczynnik trafień - procent stron znalezionych w rejestrach asocjacyjnych.
 - Dostęp do stron „trafionych”= $20+100=120$ ns
 - Dostęp do stron nie występujących w rejestrach= $20+100+100=220$ ns
- Dla współczynnika trafień = 80%:
 - E.C.D.= $0,8*120+0,2*220=140$ ns
- Dla współczynnika trafień = 98%:
 - E.C.D.= $0,98*120+0,02*220=122$ ns

- Bity ochrony - przypisane do każdej ramki. Można w ten sposób zaznaczyć strony tylko do czytania, do czytania i pisania i do wykonywania.
- Bit poprawności - jest ustawiony, jeśli dana strona jest w przestrzeni adresowej procesu (strona jest „legalna” dla procesu).
- Jeśli strona jest poza przestrzenią adresową procesu, ma ustawiony znacznik „nielegalna”.

- Współczesne systemy zezwalają na stosowanie bardzo wielkich przestrzeni adresów logicznych (2^{32} do $>2^{64}$).
- W takim przypadku tablica stron może zawierać nawet milion wpisów. Jeśli każdy wpis to 4 B, rozmiar tablicy wyniesie 4MB, na każdy proces. Tablice takie mogą być większe niż same procesy!
- Celowy jest więc podział na mniejsze tablice.
- Przy 32-bitowej przestrzeni adresowej:
 - 20-bitowy adres strony i 12-bitową odległość na stronie
 - można zastąpić przez:
 - 10-bitowy adres strony, 10-bitową odległość na tej zewnętrznej stronie i 12-bitową odległość wewnętrzną.

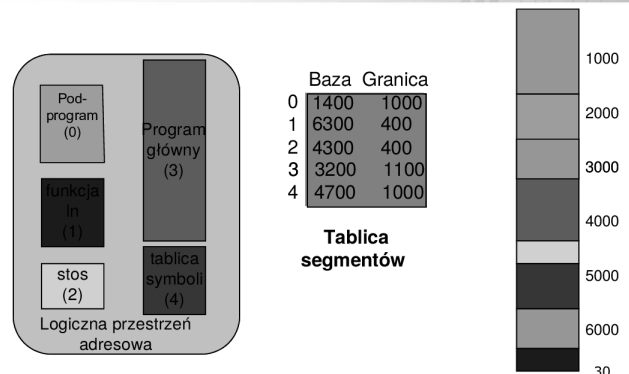
- Każdy poziom jest zapisywany jako oddzielna tablica, więc przekształcenie adresu logicznego w fizyczny może wymagać czterech dostępu do pamięci.
Czas dostępu wynosi wtedy np 520 ns.
- Ale zastosowanie pamięci podręcznej bardzo go skraca.
- Dla współczynnika trafień wynoszącego 98%:
Efektywny czas dostępu = $0,98 \cdot 120 + 0,02 \cdot 520 = 128$ ns.

- Odwrócona tablica stron ma po jednej pozycji dla każdej ramki w pamięci fizycznej
- Każda pozycja zawiera numer procesu posiadającego ramkę oraz adres wirtualny strony przechowywanej w ramce rzeczywistej pamięci.
- W systemie istnieje tylko jedna tablica stron.
- Ogranicza to zajętość pamięci, ale zwiększa czas przeszukiwania (trzeba przeszukać całą tablicę)
- Stosowanie tablic haszowania ogranicza przeszukiwanie do co najwyżej kilku wpisów.

- 20 użytkowników korzysta równocześnie z edytora tekstu. W pamięci musi się znaleźć 20 bloków danych i 20 kopii kodu edytora.
- Jeśli kod programu nie modyfikuje sam siebie w czasie działania (jest wznawialny) to można zastosować mechanizm stron dzielonych.
 - Wznawialny kod programu jest widziany przez wszystkie procesy pod tą samą lokacją.
 - Każdy proces dysponuje więc własnym obszarem danych i jednym wspólnym kodem programu.
 - Mechanizm ten może być też stosowany przy innych intensywnie używanych programach (kompilatory, systemy okien, bazy danych).
 - W systemach z odwróconą tablicą stron mechanizm ten napotyka na trudności - dlaczego?.

- Jest to mechanizm naśladujący postrzeganie pamięci tak jak użytkownik.
- Przestrzeń adresów logicznych jest zbiorem segmentów. Każdy segment ma nazwę i długość. Użytkownik określa więc każdy adres poprzez nazwę segmentu i odległość.
- Kompilator automatycznie tworzy segmenty tworząc program wynikowy. Najczęściej oddzielnymi segmentami są:
 - program główny,
 - procedury,
 - funkcje,
 - zmienne lokalne,
 - zmienne globalne,
 - stos wywołań procedur (parametry i adresy powrotu)
 - bloki wspólne (common)...

- Adres logiczny składa się z dwóch części:
numer segmentu, odległość w segmencie
- Tablica segmentów zawiera pary danych:
 - baza (fizyczny adres początku segmentu w pamięci)
 - granica (długość segmentu)
- Rejestr bazowy tablicy segmentów - adres tablicy segmentów w pamięci.
- Ponieważ programy mogą mieć bardzo różniącą się liczbę segmentów, stosuje się też rejestr długości tablicy segmentów, który służy do sprawdzenia czy podany numer segmentu jest poprawny ($< RDTs$).



Segmenty dzielone

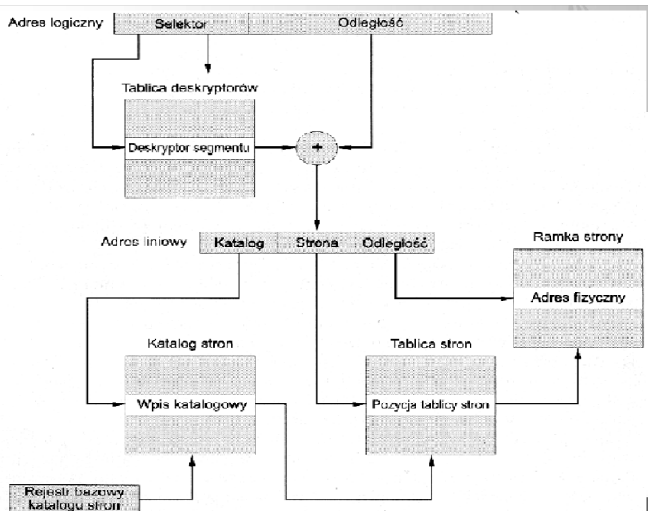
- Segmentacja ułatwia wspólne użytkowanie kodu programu, podprogramów lub niektórych danych.
- W tablicach segmentów wszystkich współużytkujących procesorów obszary (segmenty) współdzielone mają takie same wpisy bazy i granicy (wskazują więc na ten sam obszar pamięci).
- We wszystkich procesach segmenty dzielone muszą mieć ten sam numer - dlaczego?
- Alokacja segmentów w pamięci odbywa się metodą pierwszego lub najlepszego dopasowania (segmenty mają różne długości).
- Ponieważ segmentacja jest algorytmem dynamicznego przemieszczania, można przesuwać segmenty w celu lepszego upakowania.

31

Segmentacja ze stronicowaniem

- W systemie MULTICS - pozycja w tablicy segmentów nie wskazuje adresu bazowego, ale adres tablicy stron dla tego segmentu.
- Systemy oparte na Motoroli 68000 - prosta przestrzeń adresowa
- Systemy oparte na Intel (>=80386) - segmentacja ze stronicowaniem z dwupoziomym schematem stronicowania
 - maksymalna liczba segmentów w procesie: 16K
 - każdy segment mniejszy niż 4 GB
 - rozmiar strony 4 kB
 - przestrzeń adresowa ma dwie strefy po co najwyżej 8K segmentów:
 - prywatne segmenty procesorów przechowywane są w tablicy lokalnych deskryptorów,
 - wspólne segmenty procesorów przechowywane są w globalnej tablicy deskryptorów

32



Segmentacja ze stronicowaniem (Intel)

- Selektor jest 16-bitową liczbą:
 - 13 b - numer segmentu
 - 1 b - czy segment jest w lokalnej czy globalnej tablicy deskryptorów,
 - 2 b - ochrona.
- Każdy adres logiczny jest parą: selektor, odległość.
- Procesor ma 6 rejestrów segmentów (do adresowania 6 segmentów) oraz 6 rejestrów mikroprogramowych do przechowywania pozycji z lokalnej i globalnej tablicy deskryptorów,
- Sprawdzanie adresu: rejestr wyboru wskazuje na pozycję w lokalnej lub globalnej tablicy; na podstawie adresu początku segmentu i jego długości tworzy się adres liniowy (sprawdzenie poprawności); jeśli poprawny, to do bazy dodaje się odległość.

34

Stronicowanie dwupoziomowe (Intel)

- Liniowy adres jest 32-bitową liczbą:
 - 20 najstarszych bitów - numer strony,
 - 10b - wskaźnik do katalogu stron,
 - 10b - wskaźnik do tablicy stron
 - 12 najmłodszych bitów: Odległość na stronie.
- W x86_64 adresy są odpowiednio rozszerzone.

35

PAE - rozszerzenie adresu fizycznego

- 32-bitowe procesory mogą obsłużyć przestrzeń adresową:

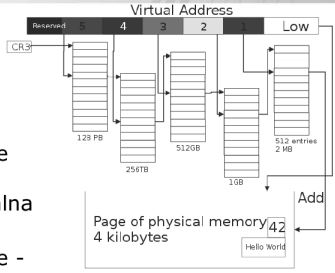
$$2^{32} = 4\,294\,967\,296 \text{ bajtów} = 4\text{GB}$$

Jak obsłużyć więcej?

- Podwajamy rozmiar wpisu w tablicy stron.
- Wskutek tego pamięci może być więcej, ale ze względu na to, że linii adresowych wciąż jest na 32 bity, **każda aplikacja może zaadresować nie więcej niż 4GB**. MMU zajmuje się tłumaczeniem przestrzeni adresowych danego programu na adresy ramek w pamięci.
- Czyli nadal widać maksimum 4GB (nie powiększymy ilości linii adresowych), ale mogą być one inne dla różnych programów.
- Rozwiązanie używane w 32-bitowych systemach Linux/Unix oraz Windows nowszych od XP.

36

- Architektura przyjęła się w typowych komputerach osobistych.
- Dwa rodzaje adresowania:
 - 4-poziomowe stronicowanie na 64-bitowych adresach. Maksymalna pamięć wirtualna to 256TB.
 - 5-poziomowe stronicowanie - pamięć wirtualna do 128PB, ale kosztem dłuższego czasu dostępu (dłuższe szukanie właściwej strony).



Dziękuję za uwagę