

Systemy Operacyjne (3)

Marcin Gogolewski
marcing@wmi.amu.edu.pl

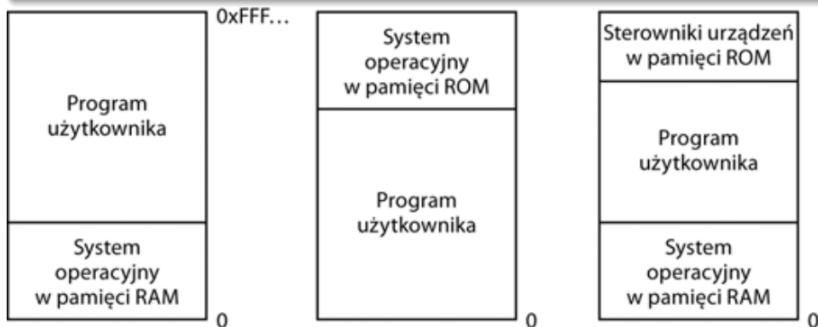
Uniwersytet im. Adama Mickiewicza w Poznaniu

Poznań, 4 listopada 2018

Abstrakcja pamięci

Problemy z adresowaniem pamięci

- jak adresować odwołania wewnątrz programu (co zrobi `MOV EAX, 1000`)?
- gdzie umieścić kod systemu operacyjnego?
- jak sobie radzić z brakiem pamięci?



Abstrakcja pamięci

Jak sobie można radzić bez abstrakcji pamięci?

- tylko jeden program jednocześnie
- wyłącznie współpracujące wątki (np. eCos)
- zabezpieczenie sprzętowe i relokacja „w locie” (jak w IBM360)

Problemy do rozwiązania

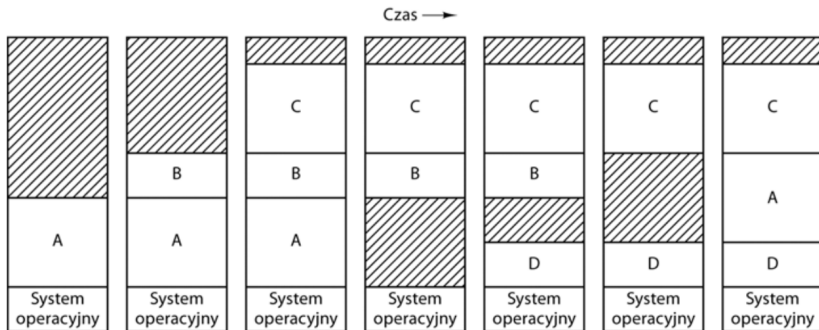
- ochrona
- relokacja

Rejestry bazy i limitu (CDC6600 i 8088)

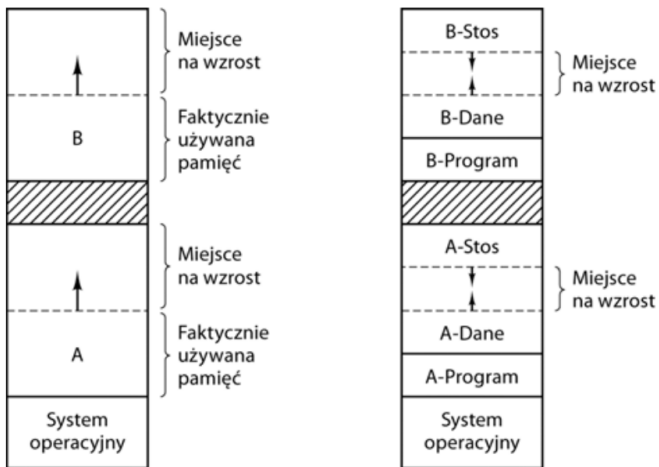
- każdy program może mieć przydzielony początek i koniec przestrzeni adresowej
- w 8088 nie było rejestru limitu. . . , tylko bazowe
- wadą wykorzystania rejestru bazowego jest konieczność wykonywania dodawania (może zająć więcej niż jeden cykl!)

Wymiana pamięci

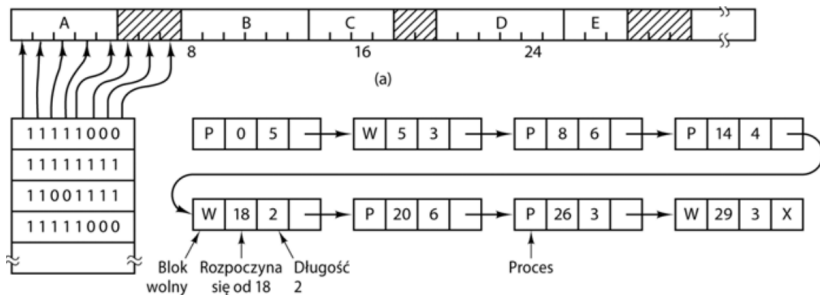
Jeżeli pamięci RAM jest za mało, można korzystać z dyskowej.



Możliwe sposoby alokacji



Zarządzanie pamięcią – mapa, czy lista



Jaki wybrać rozmiar bloku? Jakie są konsekwencje?

Pamięć wirtualna

Wcześniej

Programy zbyt duże dzielone były na „nakładki”, które trzeba było ładować (i usuwać) w odpowiedniej kolejności → trudne i prowadziło do błędów.

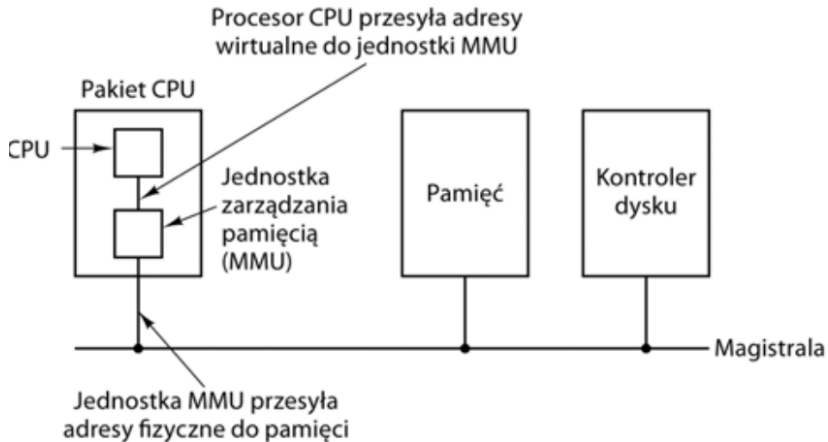
Fotheringham, 1961 – idea *pamięci wirtualnej*

Pamięć dzielona była na „strony”, które w razie potrzeby ładowane były do pamięci (a rzadko używany „wymiatane” na dysk).

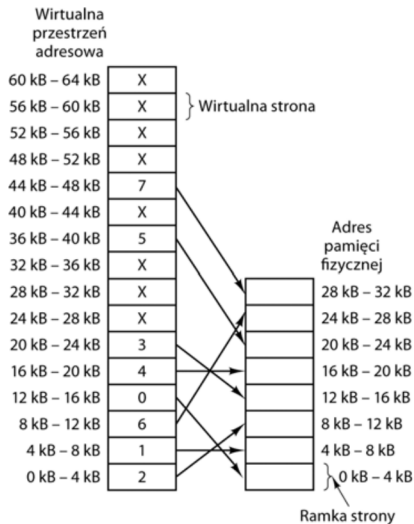
MMU – *Memory Management Unit*

Ze względu na wydajność adresy „tłumaczone” są zwykle przez dedykowany moduł sprzętowy.

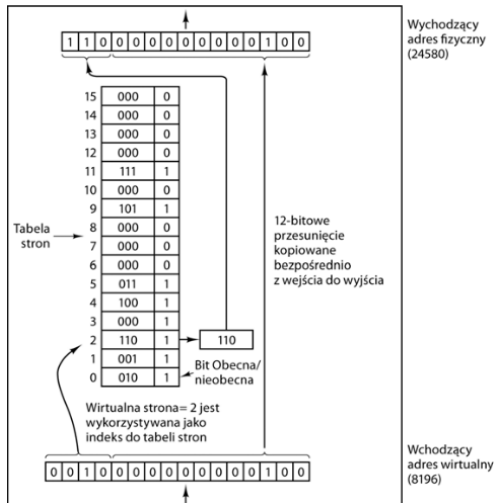
Jednostka MMU



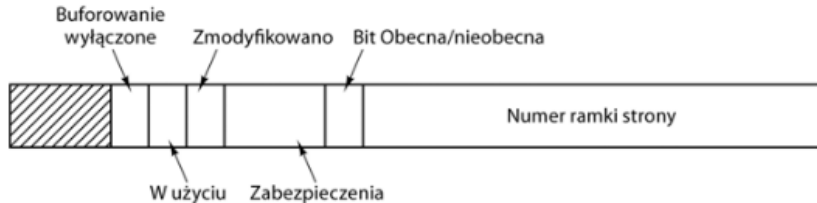
Przykładowe odwzorowanie



Wewnętrzna logika MMU



Wpis w tabeli stron



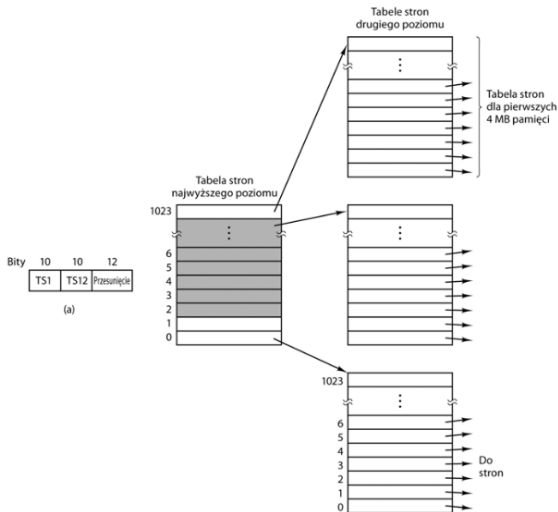
Organizacja *Translation Lookaside Buffer*

Ważność	Strona wirtualna	Zmodyfikowano	Zabezpieczenia	Ramka strony
1	140	1	RW	31
1	20	0	RX	38
1	130	1	RW	29
1	129	1	RW	62
1	19	0	RX	50
1	21	0	RX	45
1	860	1	RW	14
1	861	1	RW	75

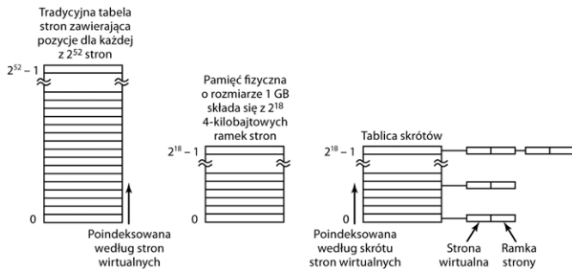
Uwaga

Teraz mogą się zdarzać „miękkie” i „twarde” błędy stron.

Wielopoziomowe tablice stron dla dużych pamięci



Tradycyjna i odwrócona tabela stron



Jaki wybrać algorytm zastępowania stron?

- NRU (Not Recently Used)
- FIFO
- algorytm drugiej szansy
- algorytm zegarowy
- LRU (Least Recently Used)
- NFU (Not Frequently Used)

Porównanie algorytmów

Algorytm	Komentarz
Optymalny	Nie można go zaimplementować, ale można go wykorzystać do porównań
NRU (ang. <i>Not Recently Used</i>)	Bardzo zgrubne przybliżenie algorytmu LRU
FIFO (ang. <i>First-In, First-Out</i>)	Może doprowadzić do usuwania ważnych stron
Drugiej szansy	Duże usprawnienie w porównaniu z FIFO
Zegarowy	Realistyczny
LRU (ang. <i>Least Recently Used</i>)	Doskonały, ale trudny do dokładnego zaimplementowania
NFU (ang. <i>Not Frequently Used</i>)	Mało dokładne przybliżenie algorytmu LRU
Postarzanie	Wydajny algorytm, który dobrze przybliży algorytm LRU
Zbiór roboczy	Dość kosztowny do zaimplementowania
WSClock	Dobry i wydajny algorytm