JĄDRO SYSTEMOWE KERNEL

WYKONAŁ:

RAFAŁ HUDASZEK

PLAN PROJEKTU

1. Jądro systemowe
   - Troche historii
   - Kernel, typy jąder i ich modele komunikacji
   - Cechy systemowe
   - Zadania i funkcje
     - Zarządzenie pamięcią
     - Stronicowanie
     - Procesy i ich ochrona

2. Porównanie Linuksa, Windows NT oraz Mach-3

TROCHĘ HISTORII

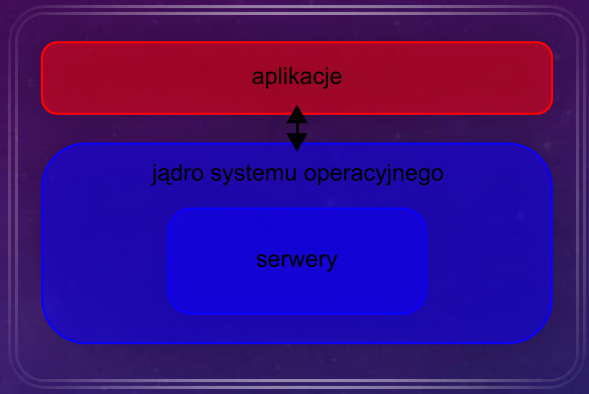
- W latach 70 UNIX rozwijał się w środowiskach i laboratoriach
- W latach 80 UNIX zaczął przenikać do biznesu
- W roku 1991 Linus Torvalds, fiński programista, ujawnił własną wersje systemu UNIX, nazwał ją "LINUX"
- Pierwszy otwarty i darmowy system operacyjny
- Linux był i jest w pełni zgodny z systemem UNIX dzięki standardom POSIX
- Posiada wiele niezależnych od siebie dystrybucji, np. Arch linux, Mint, Ubuntu i wiele innych. Wszystkie te dystrybucje mają część wspólną...

KERNEL

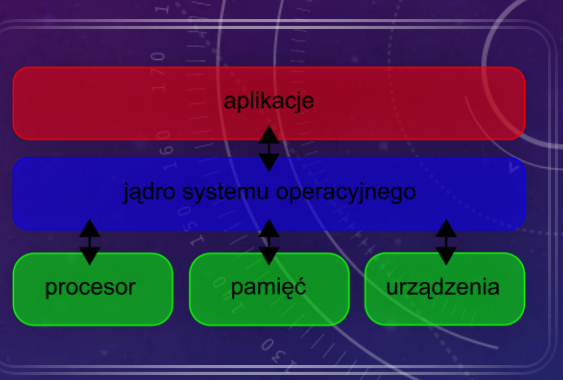
Budowa jądra i modele ich komunikacji

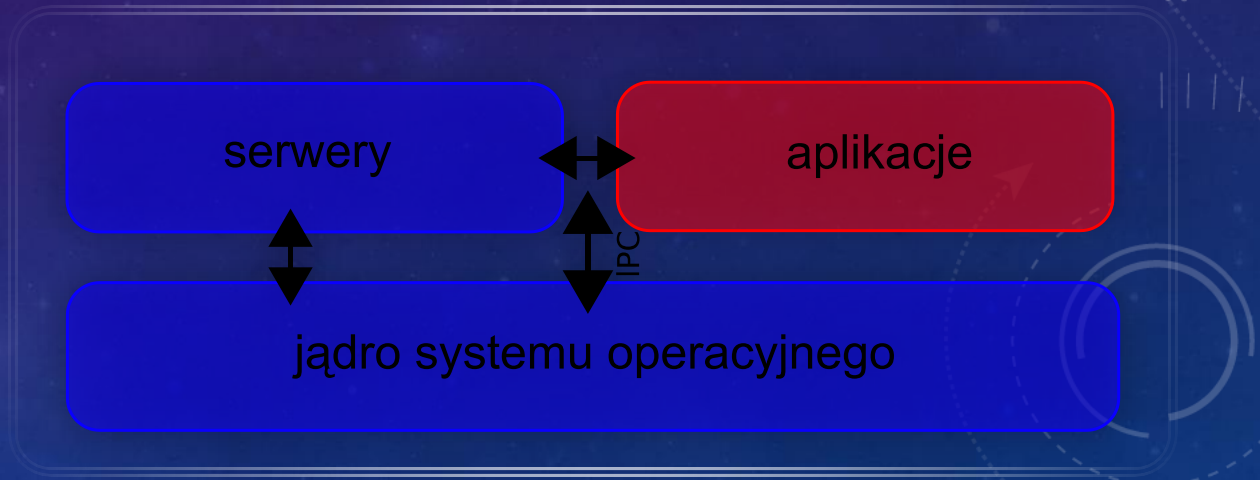
- **Jądro monolityczne**
- **Mikrojądro**
- **Nanojądro**
- **Exokernel**
- **Cachekernel**
- **Jądro hybrydowe**

Jądro hybrydowe



Jądro monolityczne





Mikrojądro

CECHY JĄDRA

Z budowy jądra wynikają jego cechy takie jak:

- Wielozadaniowość
- Wielowątkowość
- Wielobieżność
- Skalowalność
- Wywłaszczalność

ZARZADZANIE PAMIĘCIĄ PRZEZ KERNEL

Jądro systemowe wykorzystuje pamięć wirtualną, jest ona poziomem abstrakcji pomiędzy procesem żądającym dostępu do pamięci (adresowanie liniowe), a fizycznymi adresami umożliwiającymi spełnienie tych żądań. Takie rozwiązanie umożliwia:

- Działanie procesów, które wymagają więcej pamięci niż ilość pamięci RAM dostępna w systemie.
- Udostępnienie ciągłej przestrzeni adresowej, niezależnej od organizacji pamięci fizycznej.
- Stronicowanie na żądanie; w pamięci RAM przechowywana jest tylko porcja danych lub kodu, która jest obecnie używana lub wykonywana, strony nieużywane mogą być przenoszone do pamięci pomocniczej kiedy nie są potrzebne.
- I wiele wiele wiecej.

STRONICOWANIE

Stronicowanie - mapowanie adresów logicznych na adresy fizyczne

Adresem logicznym - adres widziany przez program. Zbiór wszystkich adresów logicznych tworzy tzw. logiczną **przestrzeń adresową**.

Adresem fizycznym - adres, który trafia na szynę adresową pamięci fizycznie występującej w komputerze.

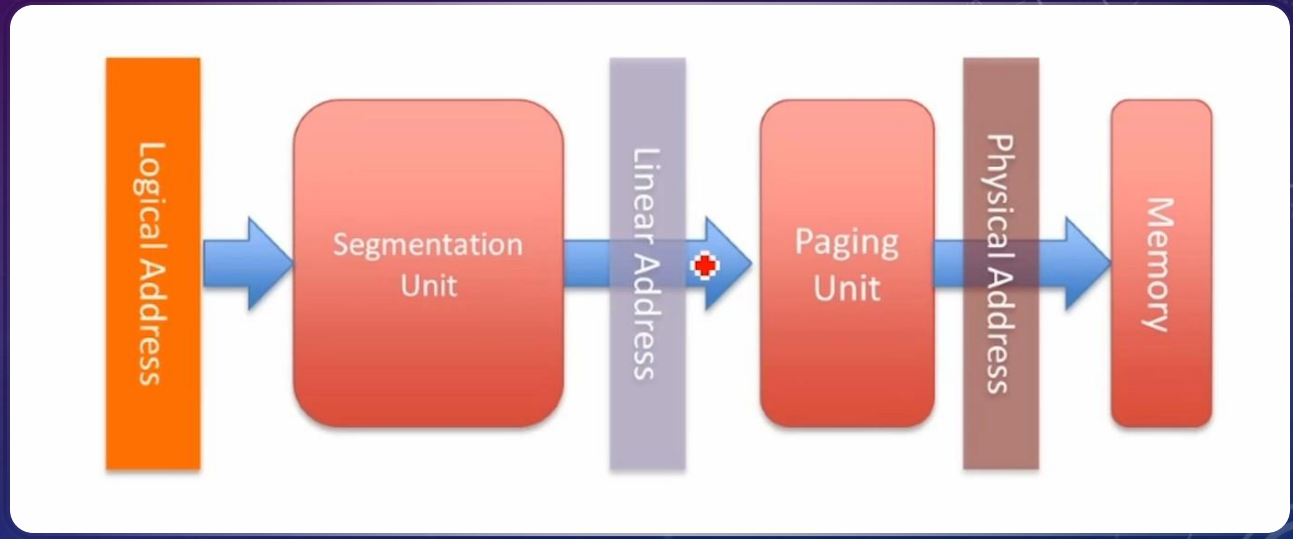
Słowo maszynowe - podstawowa porcja informacji, na której operuje system. Długość słowa w danej architekturze komputera determinuje rozmiar szyny danych oraz rejestrów procesora. Dla procesora słowem jest 16-bitowy (2-bajtowy), 32-bitowy (4-bajtowy) lub 64-bitowy (8-bajtowy) element danych

STRONICOWANIE CD.

Segmentacja - jedna z metod ochrony pamięci, używana przy wielozadaniowości

Zalety: prostota relokacji kodu i danych

Wady: nienaturalne dzielenie kodu programu

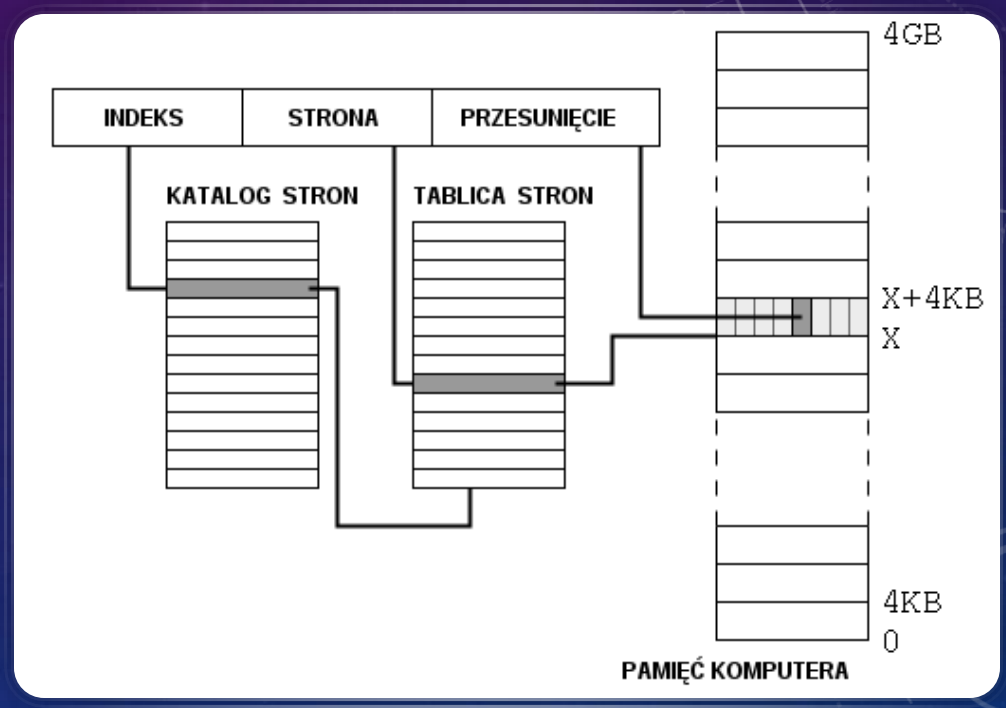


STRONICOWANIE CD.

Na przykładzie procesora intel 80386

Jeśli włączone jest stronicowanie, wówczas cała przestrzeń adresowa (4 GB) **segmentowana** jest na *strony* o rozmiarach 4 kB. Program odwołuje się do pamięci poprzez adres który jest 32-bitową liczbą, która składa się z trzech części:

- indeks w **katalogu stron** (liczba 10-bitowa),
- indeks w **tablicy stron** (liczba 10-bitowa),
- **przesunięcie** w obrębie strony (liczba 12-bitowa).

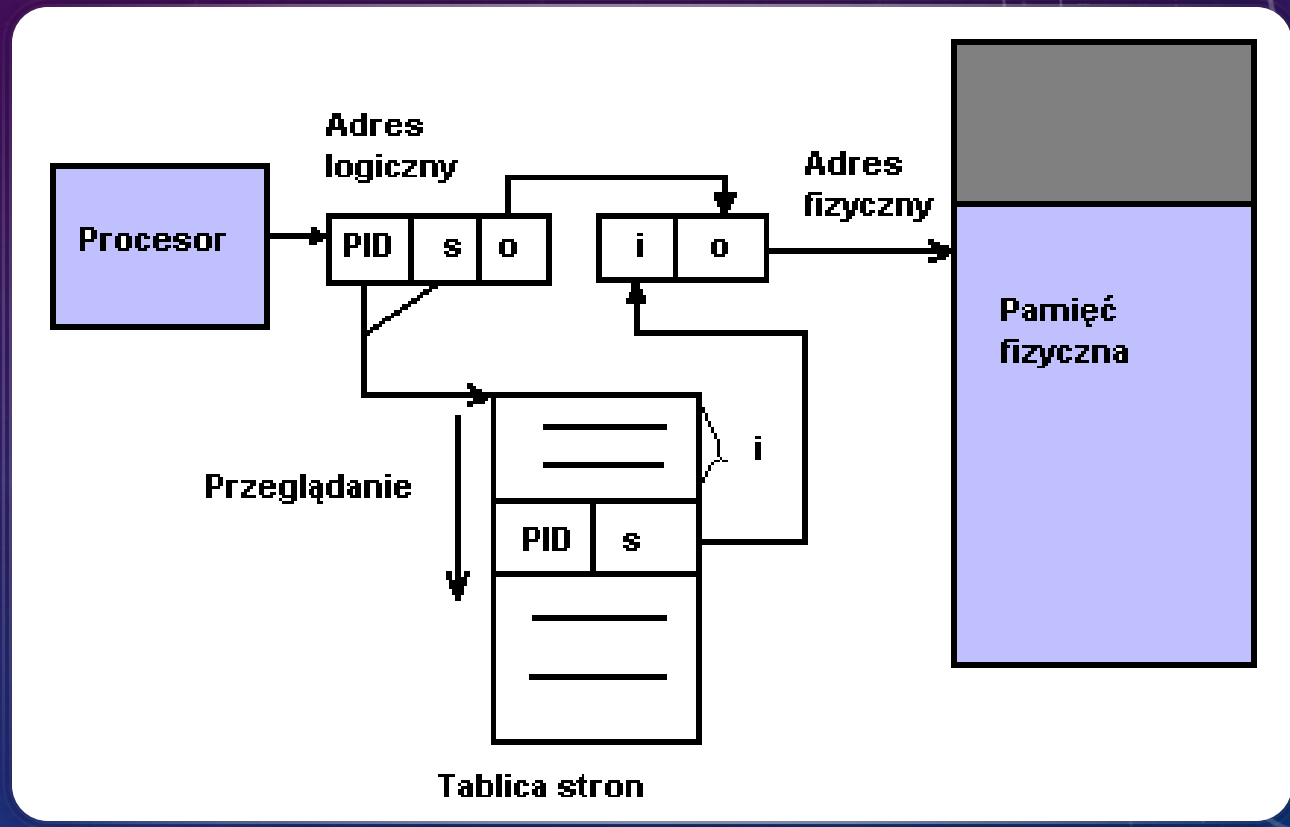


STRONICOWANIE, 64 BITOWY PROCESOR

- każdy adres wirtualny w systemie składa się z trójki:

< identyfikator-procesu, numer-strony, przesunięcie>

- każdy wpis w odwróconej tablicy stron jest parą: < identyfikator-procesu, numer-strony >
- gdy pojawi się odwołanie do pamięci, wówczas część adresu wirtualnego jest przekazywana podsystemowi pamięci
- jeśli dopasowanie się powiedzie, to tworzony jest adres fizyczny; niedopasowanie oznacza, że usiłowano użyć niedozwolonego adresu



PROCESY

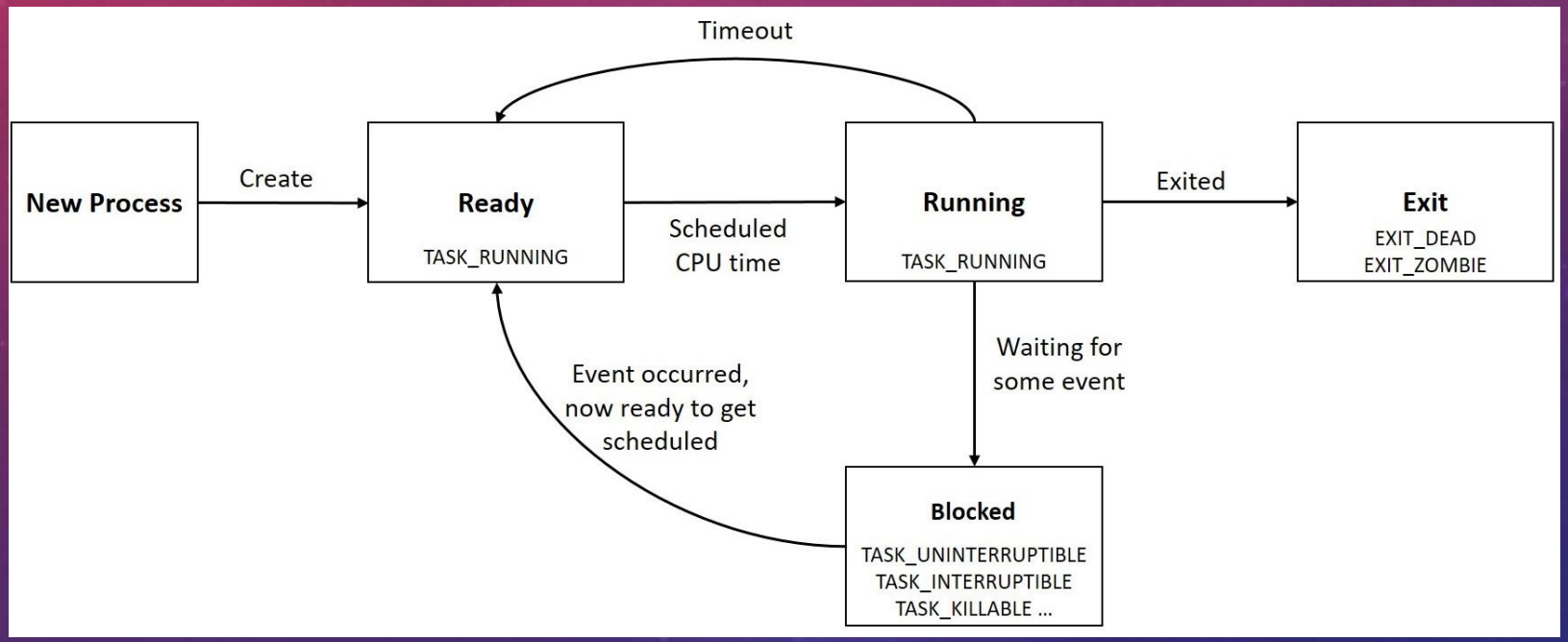
Każdy proces posiada:

- Stan
- Kontekst: zawartości wszystkich rejestrów procesora (PC (licznik rozkazów), SP (wskaźnik stosu), PSW (rejestr stanu procesora), ogólnego przeznaczenia, zarządzania pamięcią i obliczeń zmiennopozycyjnych).
- Deskryptor procesu (struct task_struct) - zawiera wszystkie informacje o procesie

Proces każdego użytkownika uruchamia się w jego własnej przestrzeni adresowej czyli przydzielonej części dostępnej, całkowitej pamięci.

- Przestrzenie adresowe (lub jej fragmenty) mogą być dzielone pomiędzy procesami na życzenie lub automatycznie jeśli kernel uzna to za stosowne.
- Oddzielenie przestrzeni adresowej procesów zapobiega ingerencji jednego procesu pamięć innego procesu, czy nawet przed ingerencją w pamięć kernela.
- Wątki jednego procesu działają we wspólnej przestrzeni adresowej lecz na oddzielnych procesorach

PROCESY CD. ORAZ KATALOG /PROC



OCHRONA PROCESÓW

Zagrożenia

1. Mało pamięci
2. Nie oddzielona przestrzeń adresowa
3. Wyścig procesów

WYŚCIG PROCESÓW, PRZYKŁAD

Wyścig – scenariusz 1

czas	Proces P1	Proces P2
1	Odczytaj wartość z pamięci (M=5) i umieść ja w rejestrze R1. (R1=5)	NOP
2	Zwiększ wartość rejestru R1 o jeden. (R1=6)	NOP
3	NOP	Odczytaj wartość z pamięci (M=5) i umieść ją w rejestrze R1. (R1=5)
4	NOP	Zmniejsz zawartość rejestru R1 o jeden. (R1=4)
5	Zapisz zawartość rejestru R1 do pamięci. (M=6)	NOP
6	NOP	Zapisz zawartość rejestru R1 do pamięci. (M=4)

Wynik Wartość wynosi 4 (jako ostatni swój wynik do pamięci zapisał proces drugi). Tymczasem prawidłowym wynikiem jest 5.

Wyścig – scenariusz 2

czas	Proces P1	Proces P2
1	Odczytaj wartość z pamięci (M=5) i umieść ja w rejestrze R1. (R1=5)	NOP
2	NOP	Odczytaj wartość z pamięci (M=5) i umieść ją w rejestrze R1. (R1=5)
3	Zwiększ wartość rejestru R1 o jeden. (R1=6)	NOP
4	NOP	Zmniejsz zawartość rejestru R1 o jeden. (R1=4)
5	NOP	Zapisz zawartość rejestru R1 do pamięci. (M=4)
6	Zapisz zawartość rejestru R1 do pamięci. (M=6)	NOP

Wynik Wartość wynosi 6 (jako ostatni swój wynik do pamięci zapisał proces pierwszy). Tymczasem prawidłowym wynikiem jest 5.

SYSTEM CALLS

Czyli interfejs między wykonywanym programem a (posiadającym zwykle wyższe uprawnienia) jądrem systemu operacyjnego.

Przykładami wywołań systemowych mogą być:

- dostęp do systemu plików,
- komunikacja międzyprocesowa,
- uruchamianie innych programów,
- sterowanie urządzeniami systemowymi,
- obsługiwanie komunikacji sieciowej.

Na ćwiczeniach laboratoryjnych jedno z zadań będzie dotyczyło wywołań systemowych

BIBLIOGRAFIA

- http://kernel.wikidot.com/kernel:teoria:podsystemy
- https://subscription.packtpub.com/book/application_development/9781785883057/1/ch01lvl1sec9/process-descriptors
- Wikipedia

DZIĘKUJEMY ZA UWAGE :)

SOMEONE@EXAMPLE.COM