

SYSTEMY OPERACYJNE

Wykład dla kierunków:

- Informatyka
- Matematyka stosowana i technologie informatyczne

PROWADZI:

Jarosław Bilski

Katedra Inteligentnych Systemów Informatycznych

d. Katedra Inżynierii Komputerowej

Politechnika Częstochowska

ZAJĘCIA:

Obowiązkowe

- Wykład 2 godziny tygodniowo
- Laboratorium 2 godziny tygodniowo

PODSTAWOWA LITERATURA:



- ✖ Abraham Silberschatz, Peter B. Galvin,
Greg Gagne
- Podstawy systemów operacyjnych, WNT 2005
- ✖ Andrew S. Tanenbaum
- Systemy operacyjne, Helion 2013
- ✖ William Stallings
- Systemy operacyjne, Struktura i zasady budowy, Mikom/PWN 2006
- ✖ M. Lister, R. D. Eager
- Wprowadzenie do systemów operacyjnych, WNT 1994

LITERATURA DODATKOWA:



- ✖ Andrew S. Tanenbaum
- Rozproszone systemy operacyjne, PWN 1997
- ✖ G. Couloris, J. Dollimore, T. Kindberg
- Systemy rozproszone, podstawy i projektowanie, WNT 1998

STRONA

kisi.pcz.pl

kik.pcz.pl

Dydaktyka-Ogólnoakademickie-Systemy operacyjne

UMIEJSZCZENIE PRZEDMIOTU



(C) KISI d.KIK PCz 2020

Systemy operacyjne

7

TEMatyKA WYKŁADÓW:

- ✖ Rodzaje systemów operacyjnych
- ✖ Zadania i właściwości systemu operacyjnego
- ✖ Procesy współbieżne
- ✖ Jądro systemu
- ✖ Zarządzanie pamięcią operacyjną
- ✖ Wejście i wyjście
- ✖ System plików

(C) KISI d.KIK PCz 2020

Systemy operacyjne

8

TEMatyKA WYKŁADÓW C.D.:

- ✖ Przydział zasobów i planowanie.
- ✖ Ochrona zasobów.
- ✖ Niezawodność systemu.
- ✖ Sposób komunikowania się użytkownika z systemem.
- ✖ Komunikacja międzymurowa.

RODZAJE SYSTEMÓW OPERACYJNYCH

(C) KISI d.KIK PCz 2020

Systemy operacyjne

9

(C) KISI d.KIK PCz 2020

Systemy operacyjne

10

PYTANIA:

- ✖ Co to jest system operacyjny?
- ✖ Co robi system operacyjny?
- ✖ Dlaczego jest potrzebny?
- ✖ Jak działa system operacyjny?

SKŁAD SYSTEMU KOMPUTEROWEGO



(C) KISI d.KIK PCz 2020

Systemy operacyjne

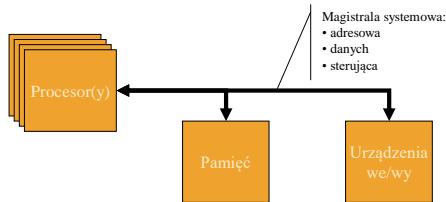
11

(C) KISI d.KIK PCz 2020

Systemy operacyjne

12

OGÓLNA BUDOWA KOMPUTERA



WYMAGANIA UŻYTKOWNIKA STAWIANE SYSTEMOWI OPERACYJNEMU

- ✖ Wygodny w użyciu
- ✖ Wydajna eksploatacja sprzętu

PODSTAWOWE CELE SYSTEMU OPERACYJNEGO

- ✖ Podział zasobów
- ✖ Tworzenie maszyny wirtualnej
- ✖ Stworzenie środowiska dla aplikacji użytkownika

Zasoby systemu to wszystkie jego elementy, które mogą być wykorzystane przez system lub aplikację użytkownika (np. pamięć, procesory, urządzenia peryferyjne, pliki, zmienne,...).

Tworzenie maszyny wirtualnej polega na takiej zmianie właściwości maszyny fizycznej, aby były one przydatniejsze dla użytkownika.

ELEMENTY MASZYNY WIRTUALNEJ VM

- ✖ Wejście-wyjście
- ✖ Pamięć operacyjna
- ✖ System plików
- ✖ Ochrona zasobów i system błędów
- ✖ Współdziałanie programów
- ✖ Sterowanie programami

WEJŚCIE-WYJŚCIE

- ✖ Stworzenie urządzeń wirtualnych
- ✖ Uproszczenie dostępu do urządzeń we-wy
- ✖ Uwolnienie użytkownika od bezpośredniego programowania urządzeń we-wy
- ✖ Ograniczenie (uniemożliwienie) bezpośredniego dostępu do urządzeń we-wy

PAMIĘĆ OPERACYJNA

- ✖ Stworzenie pamięci wirtualnej
- ✖ Zwiększenie pojemności pamięci
- ✖ Podział między użytkowników
- ✖ Ochrona pamięci
- ✖ Umożliwienie współdzielenia pamięci

(C) KISI d.KIK PCz 2020

Systemy operacyjne

19

SYSTEM PLIKÓW

- ✖ Tworzy strukturę gromadzenia plików
- ✖ Umożliwia dostęp do pamięci masowej poprzez nazwy symboliczne bez dokładnej znajomości miejsca położenia pliku

(C) KISI d.KIK PCz 2020

Systemy operacyjne

20

OCHRONA ZASOBÓW I SYSTEM BŁĘDÓW

- ✖ Ochrona przed skutkami błędów
- ✖ Ochrona przed złośliwością lub złymi zamiarami pozostałych użytkowników
- ✖ Obsługa błędów sprzętu
- ✖ Obsługa błędów użytkownika

(C) KISI d.KIK PCz 2020

Systemy operacyjne

21

WSPÓŁDZIAŁANIE PROGRAMÓW

- ✖ Współpraca w celu wykonania jednego zadania
- ✖ Wyniki uzyskane z działania jednego programu służą drugiemu
- ✖ Komunikacja między działającymi programami
- ✖ Synchronizacja działania programów

(C) KISI d.KIK PCz 2020

Systemy operacyjne

22

STEROWANIE PROGRAMAMI

Element odpowiedzialny za porozumiewanie się operatora z maszyną tzw. interface użytkownika. Występuje jako:

- ✖ język poleceń
- ✖ system graficzny
- ✖ system głosowy



Prototyp myszy z 1964 r., wykonany wg projektu Douglaasa C. Engelbart'a

(C) KISI d.KIK PCz 2020

Systemy operacyjne

23

RODZAJE SYSTEMÓW OPERACYJNYCH

- ✖ Systemy dla indywidualnego użytkownika
- ✖ Sterowanie procesami
- ✖ Systemy przepływu danych
- ✖ Przetwarzanie transakcji
- ✖ Systemy ogólnego przeznaczenia

(C) KISI d.KIK PCz 2020

Systemy operacyjne

24

SYSTEMY DLA INDYWIDUALNEGO UŻYTKOWNIKA

- ✖ wygoda i prostota
- ✖ jednozadaniowe: CP/M, MS DOS
- ✖ wielozadaniowe: OS/2, Windows NT, 2000, XP, Vista,

(C) KISI d.KIK PCz 2020

Systemy operacyjne

25

STEROWANIE PROCESAMI

- ✖ systemy czasu rzeczywistego
- ✖ Cechy: sprężenie zwrotne, niezawodność, bezpieczne wyłączenie po awarii
- ✖ Zastosowania: produkcja, kontrola, monitorowanie
- ✖ Podział: rygorystyczne, łagodne

(C) KISI d.KIK PCz 2020

Systemy operacyjne

26

SYSTEMY PRZEPTYWANIA PLIKÓW

- ✖ przeglądanie i modyfikacja dużych baz danych
- ✖ Zastosowania: systemy w urzędach, policyjne, dla lekarzy

(C) KISI d.KIK PCz 2020

Systemy operacyjne

27

SYSTEMY PRZETWARZANIA TRANSAKCJI

- ✖ bardzo częste zmiany w bazie danych
- ✖ Konieczność utrzymania integralności danych
- ✖ Zastosowania: system rezerwacji miejsc, system rozliczeń bankowych

(C) KISI d.KIK PCz 2020

Systemy operacyjne

28

SYSTEMY OGÓLNEGO PRZENACZENIA

- ✖ duża różnorodność zadań
- ✖ duża liczba programów użytkowych
- ✖ konieczność obsługi wielu urządzeń peryferyjnych
- ✖ W oparciu o nie buduje się systemy przepływu plików i przetwarzania transakcji

(C) KISI d.KIK PCz 2020

Systemy operacyjne

29

PODZIAŁ SYSTEMÓW OPERACYJNYCH

- ✖ Systemy wsadowe
zdalne wprowadzanie prac
- ✖ Systemy wielodostępne
interaktywne

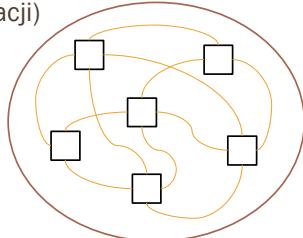
(C) KISI d.KIK PCz 2020

Systemy operacyjne

30

PODZIAŁ SYSTEMÓW OPERACYJNYCH 2

- ✗ jednostanowiskowe
- ✗ rozproszone (koordynacja zadań i przepływu informacji)



(C) KISI d.KIK PCz 2020

Systemy operacyjne

31

STRUKTURA SYSTEMU:

- ✗ jednolita
- ✗ warstwowa
- ✗ klient-serwer

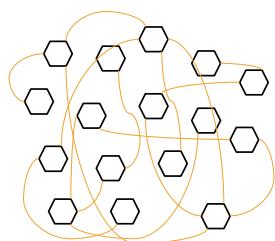
(C) KISI d.KIK PCz 2020

Systemy operacyjne

32

JEDNOLITA STRUKTURA SYSTEMU

- ✗ zespół wzajemnie wywołujących się procedur - trudnośc i konserwacji



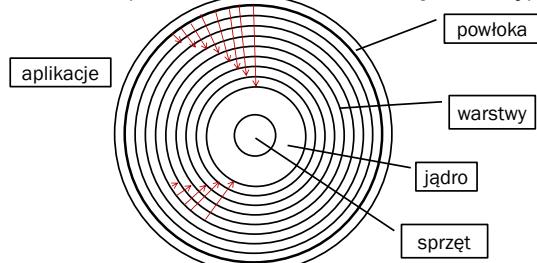
(C) KISI d.KIK PCz 2020

Systemy operacyjne

33

WARSTWOWA STRUKTURA SYSTEMU

- ✗ podział systemu na moduły o określonych zadaniach (odwołania tylko do niższej warstwy)



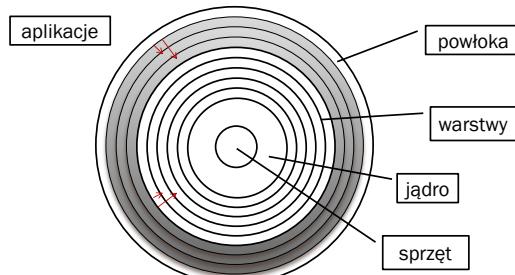
(C) KISI d.KIK PCz 2020

Systemy operacyjne

34

WARSTWOWA STRUKTURA SYSTEMU

Wersja pierścieniowa



(C) KISI d.KIK PCz 2020

Systemy operacyjne

35

STRUKTURA SYSTEMU TYPU Klient-SERWER

- ✗ moduły komunikują się poprzez wysyłanie komunikatów od klienta do serwera
- ✗ przekazywaniem komunikatów zajmuje się mikrojądro (np. Windows NT)
- ✗ łatwa rozbudowa i konserwacja

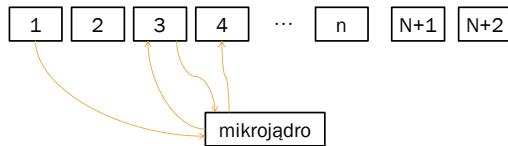
(C) KISI d.KIK PCz 2020

Systemy operacyjne

36

STRUKTURA SYSTEMU TYPU Klient-SERWER

budowa i działanie



(C) KISI d.KIK PCz 2020

Systemy operacyjne

37

SYSTEM VMS

- ✖ różne systemy
- ✖ wirtualne procesory

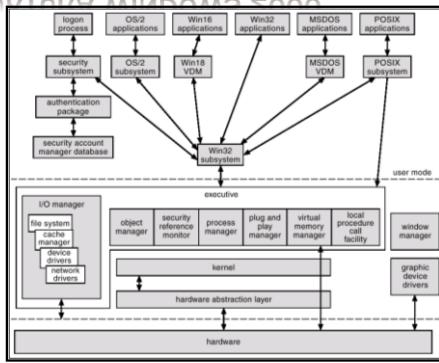
AU	AU	AU
SO	SO	SO
„Surowa Maszyna Wirtualna” - VMS		
Surowa Maszyna		

(C) KISI d.KIK PCz 2020

Systemy operacyjne

38

STRUKTURA WINDOWS 2000

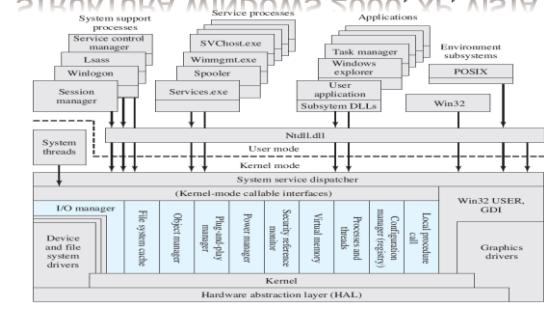


(C) KISI d.KIK PCz 2020

Systemy operacyjne

39

STRUKTURA WINDOWS 2000, XP, VISTA

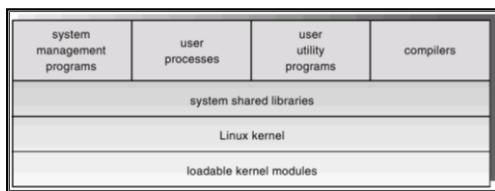


(C) KISI d.KIK PCz 2020

Systemy operacyjne

40

STRUKTURA LINUKSA

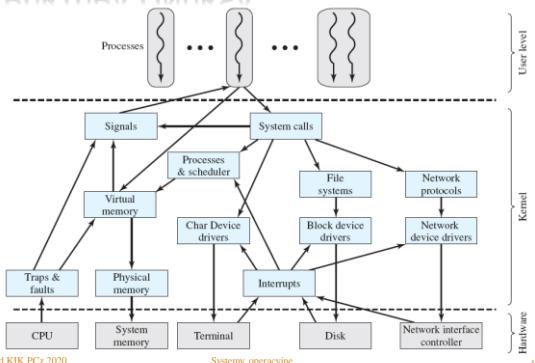


(C) KISI d.KIK PCz 2020

Systemy operacyjne

41

STRUKTURA LINUKSA

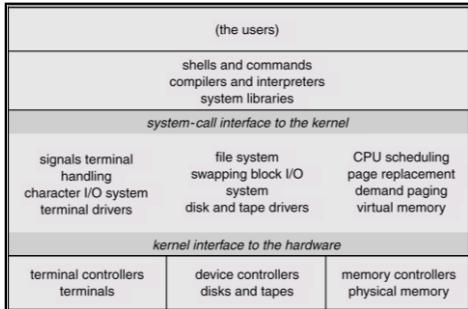


(C) KISI d.KIK PCz 2020

Systemy operacyjne

42

STRUKTURA 4.3 FREE BSD



(C) KISI d.KIK PCz 2020

Systemy operacyjne

43

WIRTUALNA MASZyna JAVY

- ✖ tworzona dla każdego systemu oddzielnie
- ✖ pkod – pseudokod, kod binarny, kod bajtowy – niezależny od systemu
- ✖ kompilator JIT (ang. Just-in-time compilation)

(C) KISI d.KIK PCz 2020

Systemy operacyjne

44

WIRTUALIZACJA

- ✖ Vmware
- ✖ Microsoft Virtual PC
- ✖ Microsoft Serwer Hyper-V
- ✖ Sun xVM VirtualBox
- ✖ Qemu
- ✖ Xen
- ✖ Bochs

(C) KISI d.KIK PCz 2020

Systemy operacyjne

45

WIRTUALIZACJA - RODZAJE

- ✖ Wirtualizacja serwerów
- ✖ Wirtualizacja aplikacji
- ✖ Wirtualizacja prezentacji
- ✖ Wirtualizacja stacji roboczych
- ✖ Wirtualizacja pamięci masowych
- ✖ Wirtualizacja stanu użytkownika
- ✖ Wirtualizacja sieci

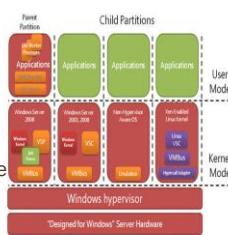
(C) KISI d.KIK PCz 2020

Systemy operacyjne

46

WIRTUALIZACJA SERWERÓW

- ✖ umożliwia aplikacjom działanie na wielu systemach operacyjnych uruchomionych na tym samym fizycznym serwerze
- ✖ możliwość pełnego wykorzystania mocy obliczeniowej i zasobów serwera, stawiając kolejne maszyny wirtualne na jednym fizycznym serwerze – zamiast kupować kolejne serwery fizyczne
- ✖ szybsze dostarczenie usług
- ✖ uproszczenie i przyspieszenie konserwacji serwerów



(C) KISI d.KIK PCz 2020

Systemy operacyjne

47

WIRTUALIZACJA APLIKACJI

- ✖ zapewnia izolację aplikacji, rozwiązuje problem zgodności różnych aplikacji, umożliwiając im działanie razem, mimo że mogą wymagać tych samych zasobów w różnych wersjach (np. różne wersje bibliotek DLL)
- ✖ usprawnia proces testowania aplikacji, nie trzeba sprawdzać, jak aplikacje współpracują ze sobą, gdy jedną z nich aktualizujemy
- ✖ aplikacje mogą być przechowywanie centralnie i strumieniowane do komputera PC w oparciu o aktualnie zalogowanego użytkownika
- ✖ całkowite usunięcie aplikacji ze stacji roboczej pozwala na zmniejszenie ilości obrazów systemów operacyjnych

(C) KISI d.KIK PCz 2020

Systemy operacyjne

48

WIRTUALIZACJA PREZENTACJI

- ✖ dostarczanie aplikacji lub całego pulpitu poprzez strumieniowanie obrazu do odbiorcy
- ✖ kod aplikacji oraz przetwarzanie danych odbywa się po stronie serwera
- ✖ umożliwia elastyczną pracę niezależnie od lokalizacji
- ✖ daje dostęp do aplikacji i całych pulpitów ze stron WWW, lokalnych komputerów w domu lub przez Internet
- ✖ upraszcza proces wdrażania aplikacji - wdrożenie jest realizowane na serwerach działających w farmie, a nie na wszystkich stacjach roboczych

WIRTUALIZACJA STACJI ROBOCZYCH

- ✖ Wirtualizacja stacji roboczych (utrzymywana na kliencie)
 - + umożliwia uruchomienie maszyn wirtualnych na jednym komputerze PC obok systemu operacyjnego hosta
 - + może być centralnie zarządzana, a obrazy maszyn wirtualnych mogą być dostarczane również centralnie
 - + zapewnienie zgodności aplikacji z systemem operacyjnym - aplikacje są zainstalowane w maszynie wirtualnej i tam wykonyuje się ich kod
 - + dostarczanie aplikacji odbywa się najczęściej w trybie seamless, czyli bez dodatkowej otoczkii pulpitu - jedynie widok samej aplikacji
- ✖ Wirtualizacja stacji roboczych (utrzymywana na serwerze)
 - + przenosi środowisko pracy, oparte na stacjach roboczych, do centrum przetwarzania danych, gdzie stacja robocza jest uruchomiona w postaci maszyny wirtualnej, a dostęp do niej zapewniający jest droga sieciowa
 - + połączenie może być zrealizowane z dowolnego komputera PC, laptopa lub cienkiego klienta
 - + nie jest jedną technologią - to połączenie wirtualizacji serwerów z wirtualizacją prezentacji.

WIRTUALIZACJA PAMIĘCI MASOWYCH

- ✖ daje niezależność lokalizacyjną dla danych przez abstrakcję fizycznej lokalizacji danych
- ✖ odwrotnie, niż w przypadku wirtualizacji serwerów, wirtualizacja pamięci masowych umożliwia, by wiele macierzy było widocznych jako jedno wirtualne urządzenie

WIRTUALIZACJA STANU UŻYTKOWNIKA

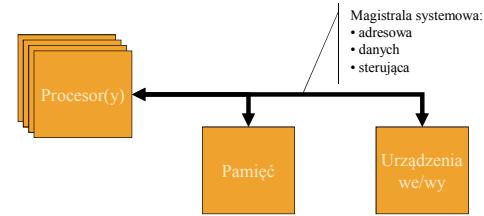
- ✖ polega na oddzieleniu danych użytkownika od komputera PC i przechowywaniu ich centralnie (np. w centrum przetwarzania danych)
- ✖ umożliwia podjęcia ich na dowolnym komputerze PC
- ✖ użytkownik ma dostarczone swoje pliki bez długiego czasu ściągania lub ręcznego kopiowania, niezależnie od miejsca logowania
- ✖ upraszcza to tworzenie kopii zapasowych danych użytkownika

WIRTUALIZACJA SIECI

- ✖ Zewnętrzna wirtualizacja sieci
 - + technologia VLAN (802.1Q), czyli logiczny podział segmentów sieci na fizycznym sprzęcie sieciowym osiągany przez znakowanie ramek
- ✖ Wewnętrzna wirtualizacja sieci
 - + tworzenie wirtualnych przełączników i portów na poziomie hiperwizora - wirtualne przełączniki dostarczają komunikację sieciową dla maszyn wirtualnych oraz łączą je z zewnętrzną fizyczną siecią

ZADANIA I WŁAŚCIWOŚCI SYSTEMU OPERACYJNEGO

SUROWA MASZYNA



(C) IISI d.KIK PCz 2013

Systemy operacyjne

1

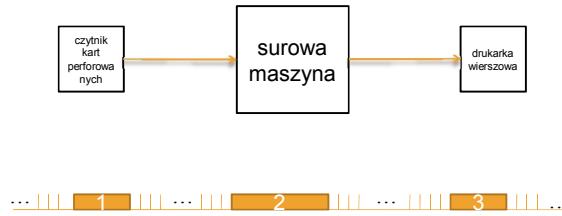
(C) IISI d.KIK PCz 2013

Systemy operacyjne

2

SYSTEM WSADOWY

- „ręczna” obsługa



(C) IISI d.KIK PCz 2013

Systemy operacyjne

3

UŁATWIENIE - JĘZYK STEROWANIA PRACAMI

- system operacyjny w wersji pierwotnej



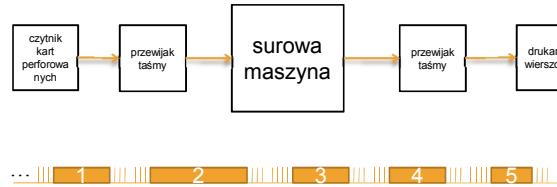
(C) IISI d.KIK PCz 2013

Systemy operacyjne

4

PRACA POŚREDNIA

- ładowanie prac przez komputer pomocniczy na taśmę magnetyczną



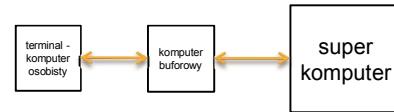
(C) IISI d.KIK PCz 2013

Systemy operacyjne

5

PRACA Z SUPERKOMPUTEREM

- ładowanie prac z terminala przez komputer buforowy

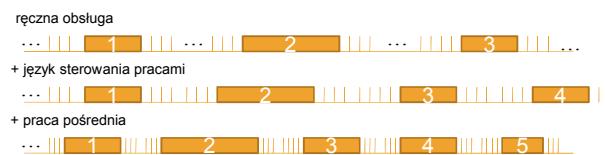


(C) IISI d.KIK PCz 2013

Systemy operacyjne

6

WYKRES ŁĄCZNY



(C) IISI d.KIK PCz 2013

Systemy operacyjne

7

MECHANIZMY SPRZĘTOWE POLEPSZAJĄCE DZIAŁANIE SUROWEJ MASZYNY

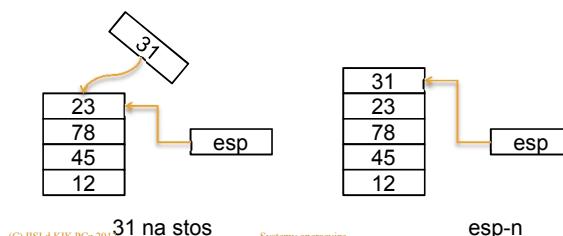
x stos

x przerwanie

x kanał

STOS

- x struktura w pamięci typu LIFO
- x należy pamiętać o właściwej kolejności pobierania

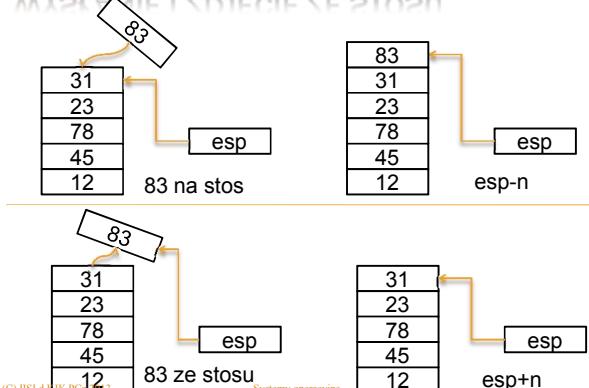


(C) IISI d.KIK PCz 2013

Systemy operacyjne

9

WYSŁANIE I ZDJĘCIE ZE STOSU



(C) IISI d.KIK PCz 2013

Systemy operacyjne

10

NA STOSIE PRZECHOWUJE SIE

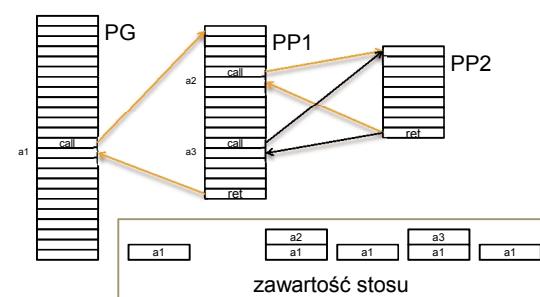
- x adres powrotu z podprogramu/przerwania
- x rejesty tymczasowo potrzebne do innych celów
- x flagi, w celu ich zabezpieczenia
- x parametry aktualne podprogramów
- x zmienne lokalne

(C) IISI d.KIK PCz 2013

Systemy operacyjne

11

WYWOŁYWANIE PODPROGRAMÓW

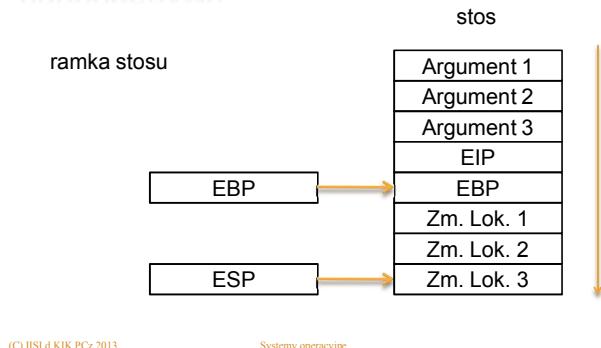


(C) IISI d.KIK PCz 2013

Systemy operacyjne

12

ZAWARTOŚĆ STOSU PO WYWOŁANIU PODPROGRAMU



(C) IISI d.KIK PCz 2013

Systemy operacyjne

13

KOD PODPROGRAMU

Parametry aktualne umieszczone są na stosie przez program wywołujący

PUSH	EBP	;Występuje jeśli Locals <> 0 lub Params <> 0
MOV	EBP,ESP	;Występuje jeśli Locals <> 0 lub Params <> 0
SUB	ESP,Locals	;Występuje jeśli Locals <> 0
 ...		
MOV	ESP,EBP	;Występuje jeśli Locals <> 0
POP	EBP	;Występuje jeśli Locals <> 0 lub Params <> 0
RET	Params	;Zawsze występuje

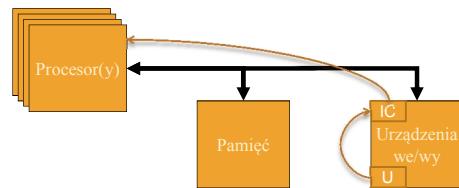
(C) IISI d.KIK PCz 2013

Systemy operacyjne

14

PRZERWANIE

- ✗ jest sygnałem od urządzenia zgłaszającym potrzebę jego obsługi
- ✗ polega na przerwaniu wykonywania programu głównego PG w celu wykonania podprogramu obsługi przerwania POP

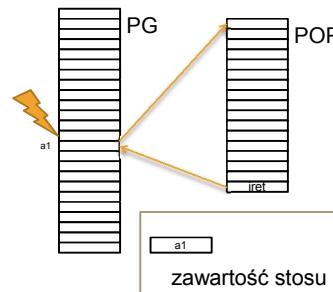


(C) IISI d.KIK PCz 2013

Systemy operacyjne

15

PRZERWANIE - DZIAŁANIE



Dla procesorów x86 na stos odkładane są rejestry: (E)IP, CS i Flags.

(C) IISI d.KIK PCz 2013

Systemy operacyjne

16

PRZERWANIA - PODZIAŁ

- ✗ maskowalne – wszystkie lub wybrane
- ✗ niemaskowalne – wykorzystywane w specjalnych sytuacjach
- ✗ z priorytetem stałym
- ✗ z priorytetem rotacyjnym

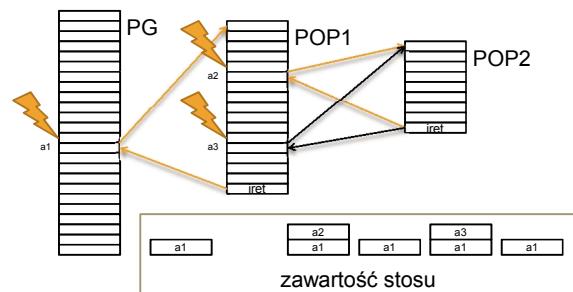
(C) IISI d.KIK PCz 2013

Systemy operacyjne

17

PRZERWANIE O WYŻSZYM PRIORYTECIE

może przerwać POP o niższym priorytecie

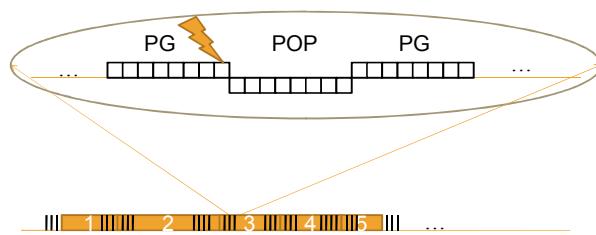


(C) IISI d.KIK PCz 2013

Systemy operacyjne

18

SYSTEM Z PRZERWANIAMI

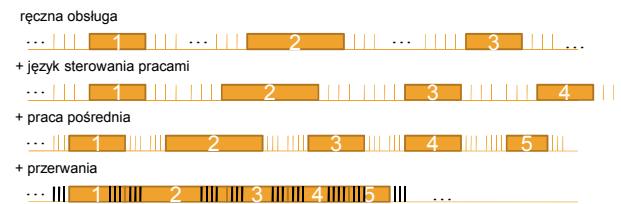


(C) IISI d.KIK PCz 2013

Systemy operacyjne

19

WYKRESY



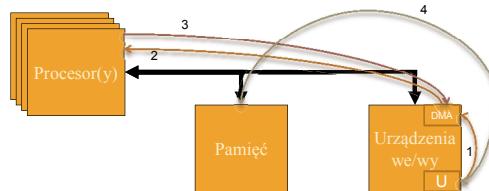
(C) IISI d.KIK PCz 2013

Systemy operacyjne

20

KANAŁ

- ✗ transmisja odbywa się z urządzenia do/z pamięci z pominięciem procesora
- ✗ procesor przechodzi w stan wysokiej impedancji i zwalnia magistralę systemową
- ✗ transmisja przez kanał kończy się przerwaniem

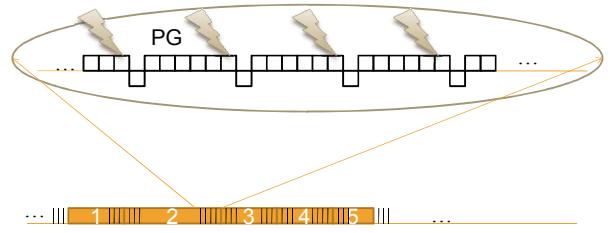


(C) IISI d.KIK PCz 2013

Systemy operacyjne

21

SYSTEM Z KANAŁEM



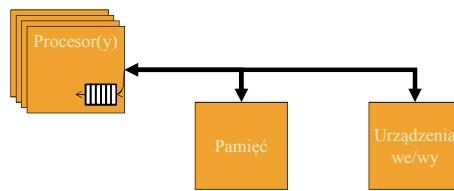
(C) IISI d.KIK PCz 2013

Systemy operacyjne

22

KOLEJKA ROZKAZÓW

Większość nowoczesnych procesorów posiada wbudowaną kolejkę rozkazów i nie musi odwoływać się do pamięci by pobrać następny rozkaz.

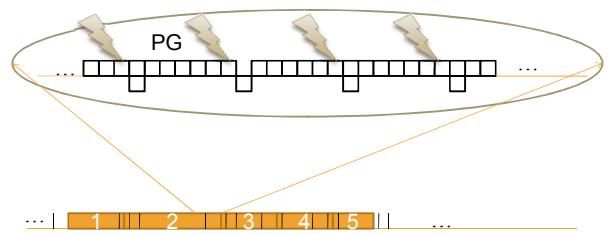


(C) IISI d.KIK PCz 2013

Systemy operacyjne

23

SYSTEM Z KANAŁEM I KOLEJKĄ ROZKAZÓW W PROCESORZE

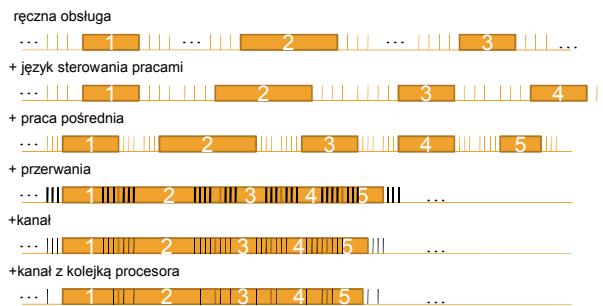


(C) IISI d.KIK PCz 2013

Systemy operacyjne

24

WYKRESY



(C) IISI d.KIK PCz 2013

Systemy operacyjne

25

MONITOR WSADOWY JEDNOSTRUMIENIOWY

- ✖ planowanie wykonywania prac z dysku



W rzeczywistości obciążenie systemu wygląda tak:



(C) IISI d.KIK PCz 2013

Systemy operacyjne

26

MONITOR WSADOWY WIELOSTRUMIENIOWY

- ✖ wieloprogramowanie – wczytywanie i uruchamianie wielu prac w tym samym czasie



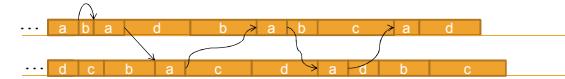
(C) IISI d.KIK PCz 2013

Systemy operacyjne

27

MONITOR WSADOWY WIELOSTRUMIENIOWY WYKORZYSTUJĄCY WIELE PROCESORÓW

- ✖ każda praca może być wykonywana na kilku procesorach
- ✖ należy pamiętać o zachowaniu i przywróceniu stanu zadania



Przełączanie prac może się odbywać w określonych przez SO odstępach czasu



(C) IISI d.KIK PCz 2013

Systemy operacyjne

28

KOLEJNE ETAPY ROZWOJU SO

- ✖ praca interakcyjna – wielodostęp
- ✖ wielosystemowość
- ✖ sieci komputerowe

(C) IISI d.KIK PCz 2013

Systemy operacyjne

29

SZCZEGÓLowe ZADANIA SYSTEMU OPERACYJNEGO:

- ✖ szeregowanie prac
- ✖ sterowanie pracami lub interpretowanie języka polecień
- ✖ obsługa błędów
- ✖ obsługa wejścia-wyjścia
- ✖ obsługa przerwań
- ✖ planowanie prac
- ✖ sterowanie zasobami
- ✖ ochrona zasobów
- ✖ umożliwienie wielodostępności
- ✖ udostępnianie dobrego sposobu komunikacji z operatorem
- ✖ ewidencjonowanie zasobów komputerowych

(C) IISI d.KIK PCz 2013

Systemy operacyjne

30

WŁAŚCIWOŚCI SYSTEMU OPERACYJNEGO:

- ✖ współprzeźność
- ✖ wspólne korzystanie z zasobów
- ✖ pamięć długookresowa
- ✖ niedeterminizm

WSPÓŁBIEŻNOŚĆ

- wykonywanie kilku czynności jednocześnie
- ✖ przełączanie między czynnościami
- ✖ wzajemna ochrona
- ✖ synchronizacja

(C) IISI d.KIK PCz 2013

Systemy operacyjne

31

(C) IISI d.KIK PCz 2013

Systemy operacyjne

32

WSPÓLNE KORZYSTANIE Z ZASOBÓW

- ✖ koszty - cały system dla jednego użytkownika to zbyt drogo
- ✖ korzystanie z pracy innych - programy i dane innych
- ✖ wspólne dane - współdzielone pliki
- ✖ oszczędność - ta sama kopia dla wielu użytkowników

PAMIĘĆ DŁUGOOKRESOWA

- ✖ z niej uruchamia się system operacyjny
- ✖ przechowuje programy i dane
- ✖ dzięki niej nie trzeba wprowadzać oprogramowania i danych przy każdym włączeniu systemu

(C) IISI d.KIK PCz 2013

Systemy operacyjne

33

(C) IISI d.KIK PCz 2013

Systemy operacyjne

34

NIEDETERMINIZM

- ✖ w sensie reakcji na różne zdarzenia w różnej kolejności
- ✖ różny ciąg przełączeń pomiędzy zadaniami

POŻĄDANE CECHY SYSTEMU OPERACYJNEGO:

- ✖ wydajność
- ✖ niezawodność
- ✖ pielęgnowalność
- ✖ mały rozmiar

(C) IISI d.KIK PCz 2013

Systemy operacyjne

35

(C) IISI d.KIK PCz 2013

Systemy operacyjne

36

WYDAJNOŚĆ

- ✗ średni czas między pracami wsadowymi
- ✗ niewykorzystany czas procesora centralnego
- ✗ długość cyklu przetwarzania prac wsadowych
- ✗ czas reagowania
- ✗ wykorzystanie zasobów
- ✗ przepustowość informacyjna

NIEZAWODNOŚĆ

- ✗ bezbłędny system operacyjny
- ✗ potrafiący reagować na wszystkie zdarzenia występujące w dowolnej kolejności

PIEŁEGNOWALNOŚĆ

- ✗ rozszerzanie systemu
- ✗ usuwanie błędów
- ✗ budowa modułowa z określona komunikacją międzymodułową
- ✗ dobra dokumentacja
 - ✗ użytkownika
 - ✗ instalatora/administratora
 - ✗ programisty
 - ✗ wewnętrzna

MAŁY ROZMIAR

- ✗ duży system operacyjny zajmuje pamięć
- ✗ jest trudniejszy w opanowaniu
- ✗ posiada potencjalnie większą liczbę błędów
- ✗ Niestety kolejne wersje systemów są coraz większe!

ROZMIAR/ZŁOŻONOŚĆ SYSTEMÓW OPERACYJNYCH

System	Ilość linii kodu (mln)
Jądro Linuksa	10
Windows 2000	35
Windows XP	40
Windows Serwer 2003	50
Windows 7	120

PROCESOR

Procesor jest elementem (jednostką) wykonującym rozkazy.

PROCESY WSPÓŁBIEŻNE

Procesor może być:

- ✖ sprzętowy - wykonuje ciąg operacji wybranych z listy rozkazów procesora. Pobiera rozkazy z pamięci operacyjnej i kolejno je wykonuje.
- ✖ sprzętowo-programowy - jest wynikiem połączenia procesora sprzętowego z oprogramowaniem. Rozkazy interpretowane są przez oprogramowanie.

PROGRAM

Program jest to statyczny zestaw instrukcji wykonywany przez jednostkę centralną (procesor) realizujący określone zadanie.

Zestawy instrukcji mogą być w formie:

- ✖ źródłowej (tekstowej) – zrozumiałej dla programisty
- ✖ binarnej (wynikowej) zrozumiałej dla maszyny.

PROCES

Proces (zadanie) jest to dynamiczny ciąg działań wykonywanych za pośrednictwem programu lub sprzętu.

- ✖ Proces można nazwać „wykonywanym programem”.
- ✖ W danej chwili na jednym procesorze tylko jeden proces jest aktywny.
- ✖ W skład procesu wchodzą: kod programu, licznik rozkazów, sekcja danych oraz stos.
- ✖ Procesowi mogą być przydzielone: procesor, pamięć, dostęp do urządzeń we/wy oraz pliki.

PROGRAM A PROCES



program



proces



procesor

PROCES

- ✖ Jeżeli program uruchamiany jest przez np. 3 użytkowników to powstaną 3 procesy.
- ✖ Jeden proces może korzystać z wielu programów.
- ✖ W skutek uruchomienia programu może powstać wiele procesów

STANY PROCESU

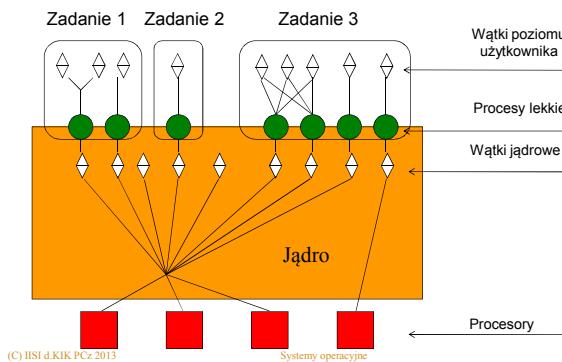
- ✖ Nowy
– proces został utworzony.
- ✖ Wykonywany (aktywny, bieżący)
– instrukcje procesu są aktualnie wykonywane
- ✖ Wykonywalny (gotowy)
– proces czeka na przydział procesora
- ✖ Oczekujący (niewykonywalny)
– proces czeka na wystąpienie jakiegoś zdarzenia.
- ✖ Zakończony
– proces zakończył działanie.

WĄTEK

Wątek to proces lekki – wątki jednego zadania współdzielą kod i dane.

- ✖ Podstawowa cecha różniąca wątek od procesu:
każdy proces posiada własną przestrzeń adresową,
natomiast wątki posiadają wspólną sekcję danych oraz kod,
ale różne stosy i liczniki rozkazów.
- ✖ Wątek jest podstawową jednostką wykorzystania procesora.
- ✖ Proces tradycyjny (nazywany ciężkim) jest równoważny zadaniu z jednym wątkiem.

WĄTKI W SYSTEMIE SOLARIS



WĄTKI I PROCESY W SYSTEMIE SOLARIS - BUDOWA

- ✖ Wątek jądrowy złożony jest z danych i stosu. Przełączanie wątków jądrowych jest stosunkowo szybkie.
- ✖ Proces lekki (LWP) zawiera blok kontrolny procesu z danymi rejestrówymi, informacjami rozliczeniowymi i informacjami dotyczącymi pamięci. Przełączanie procesów jest dość wolne.
- ✖ Wątek poziomu użytkownika wymaga tylko stosu i licznika rozkazów nie są mu potrzebne zasoby jądra, więc ich przełączanie jest szybkie. Bez względu na ilość wątków poziomu użytkownika dla jądra widoczne będą tylko procesy lekkie zadania.

WĄTKI I PROCESY W SYSTEMIE SOLARIS - PODSUMOWANIE

- ✖ Dowolne zadanie może mieć wiele wątków poziomu użytkownika.
- ✖ Wątki te mogą być planowane i przełączane bez interwencji jądra.
- ✖ Zablokowanie jednego z wątków użytkownika i podjęcie działania przez inny wątek nie wymaga przełączania kontekstu.
- ✖ Każdy proces LWP jest przyłączony do jednego wątku jądrowego,
- ✖ Wszystkie wątki poziomu użytkownika są od jądra niezależne.
- ✖ W zadaniu może być wiele procesów LWP, lecz są one używane tylko do komunikacji z jądem.
- ✖ Gdy jeden proces lekki w zadaniu zostanie zablokowany, inne mogą kontynuować działanie w ramach zadania

WSPÓŁBIEŻNOŚĆ

Współbieżność polega na wykonywaniu wielu procesów jednocześnie.

- ✖ Należy zadbać o to by jedno nie zadanie miało negatywnego wpływu na inne zadania.

RODZAJE WSPÓŁBIEŻNOŚCI

- ✖ Współbieżność pełna.
- ✖ Współbieżność pozorna.
- ✖ Współbieżność koleżeńska
- ✖ Współbieżność z wywłaszczeniem

WSPÓŁBIEŻNOŚĆ PEŁNA

- ✖ Każdy proces jest wykonywany do końca przez jeden procesor
- ✖ Liczba procesów jest mniejsza bądź równa liczbie procesorów.

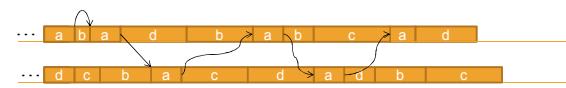
WSPÓŁBIEŻNOŚĆ POZORNA

- ✖ Występuje wówczas, gdy liczba procesów przewyższa liczbę procesorów, czyli liczba czynności jest większa od liczby wykonawców.
- ✖ System będzie dokonywał przełączeń pomiędzy procesami
- ✖ Konieczne jest zapamiętanie stanu procesu przy przełączaniu.



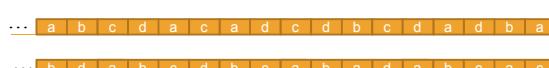
WSPÓŁBIEŻNOŚĆ KOLEŻEŃSKA

- ✖ Dany proces sam zwraca sterowanie do systemu operacyjnego, aby mógł on wykonać zadanie kolejne.
- ✖ Proces może zatem sam zarządzać czasem procesora.



WSPÓŁBIEŻNOŚĆ Z WYWŁASZCZENIEM

- ✖ Procesy same nie zwracają sterowania.
- ✖ System wywłaszcza proces z procesora.
- ✖ System operacyjny przekazuje sterowanie kolejnemu procesowi bez względu na proces poprzedni.



KOMUNIKACJA MIĘDZY PROCESAMI

Procesy:

- ✖ współpracują w celu wykonania zadania zleconego przez użytkownika
- ✖ współzawodniczą o zasoby systemu

KATEGORIE RODZAJÓW KOMUNIKACJI:

- ✖ wzajemne wyłączanie
- ✖ synchronizacja
- ✖ zakleszczenie

WZAJEMNE WYŁĄCZENIE

Polega na zapewnieniu, aby tylko jeden proces naraz mógł korzystać z zasobów niepodzielnych.

ZASOBY DZIELI SIĘ NA:

- ✖ zasoby podzielne
- ✖ zasoby niepodzielne

ZASOBY PODZIELNE

- ✖ kilka procesów może korzystać z nich współbieżnie
- ✖ zabranie zasobu przez jakiś proces inny odbywa się bez żadnej szkody.
- ✖ Do zasobów podzielnych należą:
 - + jednostki centralne
 - + pliki przeznaczone wyłącznie do odczytu
 - + pamięć wyłącznie do odczytu (kod lub stałe)

ZASOBY NIEPODZIELNE

- ✖ mogą być wykorzystywane tylko przez jeden proces
- ✖ Do zasobów niepodzielnych należą:
 - + większość urządzeń zewnętrznych
 - + pliki otwarte do zapisu
 - + pamięć przeznaczona do zapisu

NIEPODZIELNOŚĆ ZASOBÓW WYNIKA Z

- ✖ natury fizycznej danego zasobu, która nie pozwala na jego współdzielanie. Typowym przykładem jest drukarka.
- ✖ tego, że czynności jednego procesu mogą zakłócać wykonywanie innego procesu.

WZAJEMNE WYŁĄCZANIE

Gdy proces wykonuje operacje na zasobach niepodzielnych, żaden inny proces nie może wykonywać operacji na tych zasobach.

Fragment kodu, w którym proces odwołuje się do zasobów niepodzielnych, nazwany jest **sekcją krytyczną**.

(C) IISI d.KIK PCz 2013

Systemy operacyjne

31

WZAJEMNE WYŁĄCZANIE

Ograniczenie dostępu do zasobów niepodzielnych przez niewspółbieżne wykonanie sekcji krytycznej jest realizowane następująco:

*czekaj(s)
sekcja krytyczna
sygnalizuj(s)*

przy założeniu, że wartość początkowa s=1

(C) IISI d.KIK PCz 2013

Systemy operacyjne

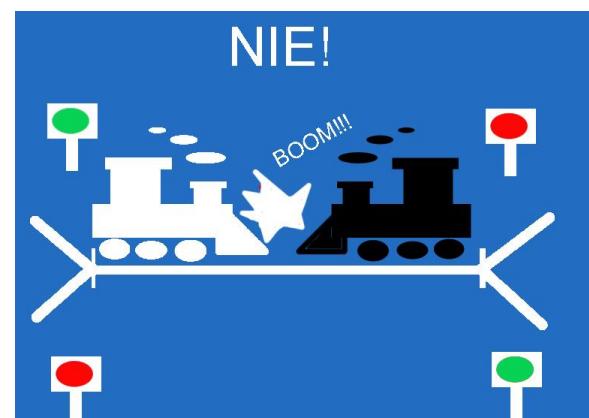
32



(C) IISI d.KIK PCz 2013

Systemy operacyjne

33



(C) IISI d.KIK PCz 2013

Systemy operacyjne

34

WZAJEMNE WYŁĄCZANIE

Jeżeli do zasobu niepodzielnego może odwoływać się kilka procesów jednocześnie (np. pula drukarek), to wartość początkową semafora ustawia się na żądaną.

(C) IISI d.KIK PCz 2013

Systemy operacyjne

35

SYNCHRONIZACJA

proces A

czekaj(s)

proces B

sygnalizuj(s)

Proces A czeka na proces B.
Proces A jest synchronizowany z procesem B.
Wartość początkowa semafora s=0.

(C) IISI d.KIK PCz 2013

Systemy operacyjne

36

SYNCHRONIZACJA

```

proces A           proces B
czekaj(s1)        sygnalizuj(s1)
                  czekaj(s2)
sygnalizuj(s2)
  
```

Procesy synchronizują się wzajemnie.
Wartość początkowa semaforów s1 i s2 = 0.

(C) IISI d.KIK PCz 2013

Systemy operacyjne

37

PRZYKŁAD

Problem producenta i konsumenta

Proces producenta wytwarza jakieś informacje, umieszcza je w buforze N.

Proces konsumenta pobiera dane z bufora i je „konsumuje”.

(C) IISI d.KIK PCz 2013

Systemy operacyjne

38

PROCES PRODUCENTA

```

repeat nieskończanie
begin
  wytwórz element;
  czekaj(miejsce dostępne);
  czekaj(manipulowanie buforem);
  umieść element w buforze;
  sygnalizuj(manipulowanie buforem);
  sygnalizuj(element dostępny);
end;
  
```

(C) IISI d.KIK PCz 2013

Systemy operacyjne

39

PROCES KONSUMENTA

```

repeat nieskończanie
begin
  czekaj(element dostępny);
  czekaj(manipulowanie buforem);
  pobierz element z bufora;
  sygnalizuj(manipulowanie buforem);
  sygnalizuj(miejsce dostępne);
  użyj element;
end;
  
```

(C) IISI d.KIK PCz 2013

Systemy operacyjne

40

PROBLEM PRODUCENTA I KONSUMENTA

```

repeat nieskończanie
begin
  wytwórz element;
  czekaj(miejsce dostępne);
  czekaj(manipulowanie buforem);
  umieść element w buforze;
  sygnalizuj(manipulowanie buforem);
  sygnalizuj(element dostępny);
end;
  
```

```

repeat nieskończanie
begin
  czekaj(element dostępny);
  czekaj(manipulowanie buforem);
  pobierz element z bufora;
  sygnalizuj(manipulowanie buforem);
  sygnalizuj(miejsce dostępne);
  użyj element;
end;
  
```

Wartości początkowe semaforów:

miejsce dostępne=rozmiar bufora
element dostępny=0
manipulowanie buforem=1

(C) IISI d.KIK PCz 2013

Systemy operacyjne

41

PROBLEM PISARZY I CZYTELNIKÓW

Dowolna liczba pisarzy i czytelników.

Gdy pisarz pisze, to inny pisarz nie może pisać ani żaden czytelnik nie może czytać.

Gdy czytelnik czyta, to inni czytelnicy mogą czytać, ale pisarze nie mogą pisać.

(C) IISI d.KIK PCz 2013

Systemy operacyjne

42

PROCES PISARZA

czekaj (s)

 pisarz pisze

sygnalizuj (s)

Wartość początkowa semafora s=1.

PROCES CZYTELNIKA

czekaj (M)

 licznik:=licznik+1

if licznik = 1 then czekaj (s)

sygnalizuj (M) (zwalniany jest semafor M)

Odczyt

czekaj (M)

 licznik:=licznik-1

if licznik=0 then sygnalizuj (s)

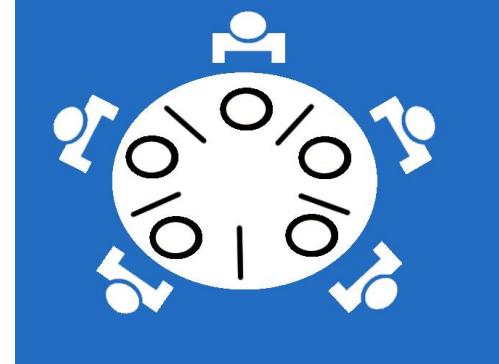
sygnalizuj (M)

Wartość początkowa semaforów s=1, M=1.

PROBLEM PIĘCIU GŁODNYCH FILOZOFÓW

- ✗ Filozofowie myślą i jedzą na zmianę.
- ✗ Każdy ma talerz i po obu stronach ma pałeczkę (jest tylko 5 pałeczek i każdy z filozofów może jeść tylko dwoma pałeczkami).
- ✗ Każdy z procesów (filozof) by działał musi wziąć 2 zasoby (pałeczki)
- ✗ Filozof żeby myśleć musi jeść.
- ✗ Co się stanie, gdy wszyscy filozofowie na raz zechcą jeść?

PROBLEM PIĘCIU GŁODNYCH FILOZOFÓW



PROBLEM PIĘCIU GŁODNYCH FILOZOFÓW

```
repeat
    czekaj (S[i]) (bierze pałeczkę)
    czekaj (S[i+1]mod 5) (2
    pałeczki)
    jedzenie
    sygnalizuj (S[i])
    sygnalizuj(S[i+1] mod 5)
until false
```

Jeżeli wszyscy filozofowie w tej samej chwili będą chcieli jeść (wszystkie procesy chcą skorzystać jednocześnie ze swoich zasobów), to powstanie **ZAKLESZCZENIE**.

PROBLEM PIĘCIU GŁODNYCH FILOZOFÓW

```
repeat
    czekaj (j)
    czekaj (S[i])
    czekaj (S[i+1])
    jedzenie
    sygnalizuj(S[i])
    sygnalizuj(S[i+1])
    sygnalizuj (j)
until false
```

Można ograniczyć filozofom dostęp do jedzenia wprowadzając maksymalną liczbę próbujących jeść filozofów do 4.

Nawet jeśli wartość semafora ustawia się na 4 nie będzie zjawiska zakleszczenia ponieważ procesy pobierają dopuszczalną ilość zasobów i oddają je.

ZAKLESZCZENIE

Niewłaściwe odwołanie się do semaforów s1 i s2 (o wartościach 1) prowadzi do zakleszczenia.

```
proces A           proces B
.
.
.
czekaj(s1)       czekaj(s2)
.
.
.
czekaj(s2)       czekaj(s1)
```

(C) IISI d.KIK PCz 2013

Systemy operacyjne

49

KONSTRUKCJE SYNCHRONIZACYJNE

- ✖ monitor
- ✖ region krytyczny

(C) IISI d.KIK PCz 2013

Systemy operacyjne

50

MONITOR

Monitor składa się z:

- ✖ - danych
- ✖ - procedur dostępu
- ✖ - programu inicjującego

(C) IISI d.KIK PCz 2013

Systemy operacyjne

51

MONITOR

```
type m = monitor
      deklaracja zmiennych
      procedura p1
      procedura p2
      .....
      procedura pN
      end;
begin
  inicjalizacje
end.
```

(C) IISI d.KIK PCz 2013

Systemy operacyjne

52

- ✖ Dostęp do zmiennych jest ograniczony.
- ✖ Można się do nich odwoływać wyłącznie przez procedury .
- ✖ W procedurach kompilator automatycznie umieszcza odwołania do niejawnie zadeklarowanych semaforów.
- ✖ Deklaracja monitora przypomina deklarację obiektu.

REGION KRYTYCZNY

Zadeklarowana zmienna v typu T, będzie używana wspólnie przez wiele procesów:

```
var v: shared T;
```

Zmienna v będzie dostępna tylko w obrębie instrukcji **region** o następującej postaci:

```
region v when B do S;
```

(C) IISI d.KIK PCz 2013

Systemy operacyjne

53

SEMAFORY W SYSTEMIE WINDOWS

- ✖ Create Semaphore(pSA, wo, w max, pN) : HANDLE
- ✖ Tworzy semafor o określonej nazwie. Zwraca uchwyt do niego.
 - ✖ pSA – wskaźnik do struktury
 - ✖ wo – wartość początkowa
 - ✖ w max – wart. max semafora (ograniczenie od góry)
 - ✖ pN – wskaźnik do nazwy semafora

SEMAFORY W SYSTEMIE WINDOWS

- ✖ Open Semaphore(at, if, pN) : HANDLE
- ✖ Zwraca uchwyty do semafora
- ✖ at – flaga dostępu do semafora
- ✖ if – mówi czy może być dziedziczony
- ✖ pN – wskaźnik do nazwy semafora

SEMAFORY W SYSTEMIE WINDOWS

- ✖ Release Semaphore(HANDLE, Const, pV) : BOOL
- ✖ Operacja sygnalizacji, zwiększa wartość semafora, którego wartość zostanie podana jako parametr ; musimy określić krok o ile będzie zmieniany (liczba całkowita). Podajemy liczbę o jaką możemy zwiększyć, niekoniecznie o 1.
- ✖ pV – wskaźnik do miejsca w pamięci, gdzie została umieszczona poprzednia wartość semafora (przed zwiększeniem)

SEMAFORY W SYSTEMIE WINDOWS

- ✖ Wait For Single Object(HANDLE, tms) : DWORD
- ✖ Czekaj na pojedynczy semafor (obiekt). Podajemy jako parametr uchwyty do semafora, a drugim parametrem jest czas oczekiwania. Jeśli czas zostanie przekroczyony to operacja zostanie zakończona.
- ✖ WAIT_OBJECT_0 – semafor jest w stanie signaled, operacja zakończona powodzeniem
- ✖ WAIT_ABANDONED - gdy obiekt przestanie istnieć
- ✖ WAIT_TIMEOUT - jeśli czas oczekiwania zostanie przekroczyony

SEMAFORY W SYSTEMIE WINDOWS

- ✖ Close Handle (HANDLE) : BOOL
- ✖ Funkcja usuwa pojedynczy semafor. Parametrem jest uchwyty do semafora. Funkcja ta usuwa też inne obiekty.

PRZEKAZYWANIE KOMUNIKATÓW

W skład narzędzi komunikacji międzyprocesowej wchodzą dwie podstawowe operacje:

- ✖ nadaj (komunikat)
- ✖ odbierz (komunikat).
- ✖ Długość komunikatu może być stała lub zmienna.
- ✖ Komunikaty mogą być kierowane do jednego procesu lub do wielu (rozgłoszeniowe).
- ✖ Można żądać potwierdzenia otrzymania komunikatu.
- ✖ Obie powyższe operacje mogą mieć różne postacie.

PRZEKAZYWANIE KOMUNIKATÓW

operacja blokująca - czeka, aż proces odbierający przyjmie komunikat.

operacja nieblokująca - po prostu umieszcza komunikat w pewnego rodzaju kolejce i pozwala, aby nadawca działał w dalszym ciągu.

PRZEKAZYWANIE KOMUNIKATÓW

Komunikacja bezpośrednia - każdy proces, który chce się komunikować, musi jawnie nazwać odbiorcę lub nadawcę uczestniczącego w tej wymianie informacji.

nadaj(P, komunikat)-najdaj komunikat do procesu P

odbierz(Q, komunikat)-odbierz komunikat od procesu Q

PRZEKAZYWANIE KOMUNIKATÓW

Komunikację bezpośrednią można przedstawić na przykładzie producenta i konsumenta.

Proces producenta:

```
repeat
  ...
  wytwarzaj jednostkę w nastp
  ...
  nadaj (konsument, nastp);
until false;
```

Proces konsumenta:

```
repeat
  odbierz (producent, nastk);
  ...
  konsumuj jednostkę z nastk
  until false;
```

PRZEKAZYWANIE KOMUNIKATÓW

Komunikacja pośrednia - komunikaty są nadawane i odbierane za pośrednictwem *skrzynek pocztowych*, nazywanych także *portami*. Możliwość komunikacji między dwoma procesami istnieje tylko wtedy, gdy mają one jakąś wspólną skrzynkę pocztową.

nadaj(A, komunikat)-najdaj komunikat do skrzynki A

odbierz(A, komunikat)-odbierz komunikat ze skrzynki A

PRZEKAZYWANIE KOMUNIKATÓW

Buforowanie komunikatów - bufor ma pewną pojemność określającą liczbę komunikatów, które mogą w nim czasowo przebywać.

- ✖ Pojemność zerowa: Maksymalna długość kolejki wynosi 0, czyli łącznie nie dopuszcza, by czekał w nim jakikolwiek komunikat. W tym przypadku nadawca musi czekać, aż odbiorca odbierze komunikat. Oba procesy muszą być zsynchronizowane.
- ✖ Pojemność ograniczona: Kolejka ma skończoną długość n, może w niej zatem pozostawać co najwyżej n komunikatów.
- ✖ Pojemność nieograniczona: Kolejka ma potencjalnie nieskończoną długość; może w niej oczekiwany dowolna liczba komunikatów. Nadawca nigdy nie jest opóźniany.

PRZEKAZYWANIE KOMUNIKATÓW

Sytuacje wyjątkowe:

- ✖ Zakończenie procesu - może się zdarzyć, że nadawca lub odbiorca zakończy działanie przed zakończeniem przetwarzania komunikatu. Pozostałą wówczas komunikaty, których nikt nigdy nie odbierze, lub jakieś procesy będą czekać na komunikaty, które nigdy nie zostaną wysłane.
- ✖ Utrata komunikatów - komunikat nadany przez proces P do procesu Q może zaginąć w sieci komunikacyjnej z powodu awarii sprzętu lub linii komunikacyjnej.
- ✖ Zniekształcenia komunikatów - komunikat może dojść do celu zniekształcony po drodze (np. wskutek zakłóceń w kanale komunikacyjnym).

STRUKTURA MSG W SYSTEMIE WINDOWS

```
typedef struct {
  HWND hwnd;
  UINT message;
  WPARAM wParam;
  LPARAM lParam;
  DWORD time;
  POINT pt;
} MSG, *PMSG;
```

Hwnd – uchwyty do okna, którego procedura okna odbiera komunikaty.
 Message – identyfikator komunikatu. Aplikacje mogą używać tylko młodszych słowa; starsze jest zarezerwowane dla systemu.
 wParam – dodatkowa informacja o komunikacie. Znaczenie zależy od komunikatu.
 lParam – dodatkowa informacja o komunikacie. Znaczenie zależy od komunikatu.
 Time – czas wysłania komunikatu.
 Pt – pozycja kurSORA na ekranie

KOMUNIKATY W SYSTEMIE WINDOWS

**BOOL PostMessage(HWND hWnd, UINT Msg,
WPARAM wParam, LPARAM lParam);**

Funkcja wysyła komunikat (Msg+wParam, lParam) umieszczając go w kolejce komunikatów okna hWnd, nie czekając na jego przetworzenie.

HWND_BROADCAST – komunikat jest wysyłany do wszystkich okien top-level w systemie. Komunikat nie jest przesyłany do okien potomnych.

KOMUNIKATY W SYSTEMIE WINDOWS

VOID PostQuitMessage(int nExitCode);

Funkcja wysyła komunikat WM_QUIT do wątków aplikacji, najczęściej w odpowiedzi na komunikat WM_DESTROY, argumentem jest kod zamknięcia aplikacji.

KOMUNIKATY W SYSTEMIE WINDOWS

**BOOL PostThreadMessage(DWORD idThread,
UINT Msg, WPARAM wParam, LPARAM lParam);**

Funkcja umieszcza komunikat (Msg+wParam, lParam) w kolejce komunikatów danego wątku idThread i natychmiastowo kończy działanie.

KOMUNIKATY W SYSTEMIE WINDOWS

**LRESULT SendMessage(HWND hWnd, UINT Msg,
WPARAM wParam, LPARAM lParam);**

Funkcja wysyła komunikat (Msg+wParam, lParam) do okna hWnd lub wielu okien, wywołując funkcje obsługi danego komunikatu, funkcja kończy działanie dopiero po obsłużeniu komunikatu przez docelowe okno.

KOMUNIKATY W SYSTEMIE WINDOWS

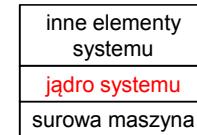
**BOOL GetMessage(LPMSG lpMsg, HWND hWnd,
UINT wMsgFilterMin, UINT wMsgFilterMax);**

Funkcja pobiera komunikat z kolejki okna o uchwycie hWnd i umieszcza go w strukturze lpMsg. Dopóki funkcja odbiera komunikaty różne od WM_QUIT, zwraca wartość większą od zera, w przeciwnym razie 0. wMsgFilterMin i wMsgFilterMax oznaczają zakres odbieranych komunikatów.

JĄDRO SYSTEMU

Jądro systemu stanowi główny interfejs między sprzętem (surową maszyną), a systemem operacyjnym.

JĄDRO SYSTEMU



PODSTAWOWE UDOGODNIENIA SPRZĘTOWE

Potrzebne w surowej maszynie do stworzenia środowiska, na którym można uruchomić bezpieczny i nowoczesny system operacyjny

- ✖ mechanizm przerwań
- ✖ ochrona pamięci operacyjnej
- ✖ zbiór rozkazów uprzywilejowanych
- ✖ zegar czasu rzeczywistego

PODSTAWOWE UDOGODNIENIA SPRZĘTOWE

Ochrona pamięci operacyjnej

- ✖ wbudowana do sprzętu adresującego pamięć
- ✖ procesy nie mogą zmienić przydzielonego im obszaru pamięci
- ✖ procesy nie mogą odwoływać się do pamięci przydzielonej innym procesom

PODSTAWOWE UDOGODNIENIA SPRZĘTOWE

Zbiór rozkazów uprzywilejowanych

- ✖ działają tylko w trybie nadzorczy

Procesor musi posiadać co najmniej 2 tryby pracy:

- ✖ nadzorcy – systemu operacyjnego, uprzywilejowany
- ✖ użytkownika - nieuprzywilejowany

PODSTAWOWE UDOGDONIENIA SPRZĘTOWE

Zbiór rozkazów uprzywilejowanych:

- ✗ włączanie i wyłączanie przerwań
- ✗ przełączanie procesora między procesami
- ✗ obsługa rejestrów pamięci wirtualnej
- ✗ operacje wejścia-wyjścia
- ✗ sterowanie procesorem

PODSTAWOWE UDOGDONIENIA SPRZĘTOWE

Przełączenie z trybu użytkownika do trybu nadzorcy:

- ✗ odwołanie do nadzorcy(systemu) - w celu wykonania rozkazu uprzywilejowanego lub funkcji systemowej
- ✗ przerwanie
- ✗ błąd w procesie użytkownika
- ✗ próba wykonania rozkazu uprzywilejowanego

PODSTAWOWE UDOGDONIENIA SPRZĘTOWE

Zegar czasu rzeczywistego - zegar sprzętowy generuje przerwania w ustalonych odstępach czasu.

Jego przerwania są podstawą

- ✗ planowania
- ✗ rozliczania zasobów
- ✗ przełączania procesów

JĄDRO SYSTEMU

Jądro systemu buduje się bezpośrednio na podstawie sprzętu, zatem jest ono najbardziej zależną od maszyny częścią systemu operacyjnego.

Jądro jest obok obsługi urządzeń peryferyjnych tą częścią systemu operacyjnego, która może zawierać kod napisany w assemblerze.

JĄDRO SYSTEMU

Zadaniem jądra jest stworzenie środowiska, w którym będą występować procesy; mieści ono w sobie

- ✗ obsługę przerwań
- ✗ przełączanie procesorów od procesu do procesu
- ✗ realizację mechanizmów służących do komunikacji między procesami

JĄDRO SYSTEMU

Jądro systemu składa się z trzech programów:

- ✗ programu wstępnej obsługi przerwań
- ✗ programu zwanego dyspozytorem
- ✗ operacji czekaj i sygnalizuj (dwóch procedur, które realizują operacje służące do komunikacji między procesami)

JĄDRO SYSTEMU

Jądro systemu zawiera również strukturę danych nazwaną **tablicą główną**. Zawiera ona szereg danych opisujących:

- ✗ **system operacyjny**
- ✗ **jego bieżący stan**
- ✗ **informacje o sprzęcie**
- ✗ **informacje o procesach**
- ✗ ...

JĄDRO SYSTEMU

Tablica główna

Nazwa SO
Wersja SO
Data i czas uruchomienia
Bieżąca data i czas
Model procesora
Ilość pamięci
Struktura procesów
Inne
...

DESKRYPTOR PROCESU

Każdy proces jest reprezentowany w systemie przez strukturę nazywaną deskryptorem procesu lub:

- ✗ blokiem kontrolnym procesu
- ✗ wektorem stanu

Jest on strukturą zawierającą wszystkie istotne informacje o procesie np.:

- ✗ identyfikator procesu (nazwę procesu)
- ✗ identyfikator stanu procesu
- ✗ pełę informacji o procesie, które powinny być zapamiętane wtedy, gdy procesor wstrzymuje jego wykonywanie.

DESKRYPTOR PROCESU

Deskryptor procesu

Identyfikator
Nazwa
Data i czas uruchomienia
Czas działania
Użytkownik
Priorytet
Stan
Inne
Środowisko ulotne
Następny w strukturze
Następny w kolejce

STANY PROCESU

- ✗ Nowy
 - proces został utworzony.
- ✗ Wykonywany (aktywny, bieżący)
 - instrukcje procesu są aktualnie wykonywane
- ✗ Wykonywalny (gotowy)
 - proces czeka na przydział procesora
- ✗ Oczekujący (niewykonywalny)
 - proces czeka na wystąpienie jakiegoś zdarzenia.
- ✗ Zakończony
 - proces zakończył działanie.

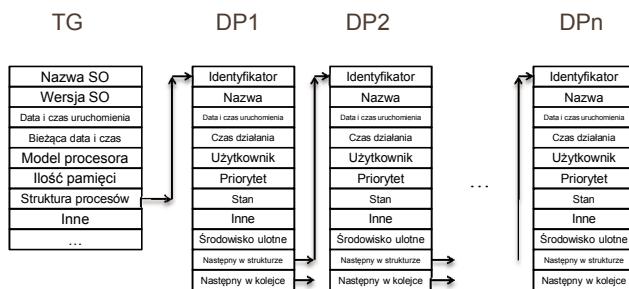
DESKRYPTOR PROCESU

Środowisko ulotne (kontekst, deskryptor sprzętowy) procesu definiuje się jako zbiór danych, które muszą być zapamiętane podczas wstrzymywania procesu, a odtworzone podczas wznowiania, aby proces działał poprawnie.

Do tych informacji należą:

- ✗ wartości wszystkich rejestrów procesora
- ✗ wartości rejestrów stosowanych do adresowania pamięci operacyjnej

TABLICA GŁÓWNA



(C) IISI d.KIK PCz 2013

Systemy operacyjne

19

(C) IISI d.KIK PCz 2013

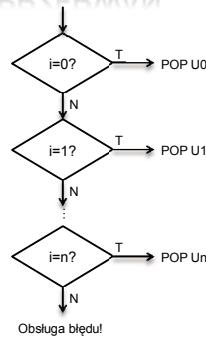
Systemy operacyjne

20

WSTĘPNA OBSŁUGA PRZERWAŃ

W systemie z tylko jednym przerwaniem należy wykonać ciąg testów, nazywany łańcuchem przeskórek.

Oczywistą korzyść przyniesie takie ułożenie łańcucha, by najczęstsze źródła przerwań znalazły się na jego początku.



(C) IISI d.KIK PCz 2013

Systemy operacyjne

21

(C) IISI d.KIK PCz 2013

Systemy operacyjne

22

WSTĘPNA OBSŁUGA PRZERWAŃ

W systemach, w których ilość linii przerwań jest mniejsza od ilości urządzeń zgłaszających przerwanie, wstępna obsługa przerwań jest grupowana tylko dla przerwań wspólnych dla kilku urządzeń.

WSTĘPNA OBSŁUGA PRZERWAŃ

W systemach umożliwiających obsługę wielu przerwań oddzielnie wstępna obsługa przerwań jest zbędna.

✗ źródło przerwania jest określone sprzętowo na podstawie sygnału przerwania

✗ obsługa przerwania jest inicjowana przez procesor

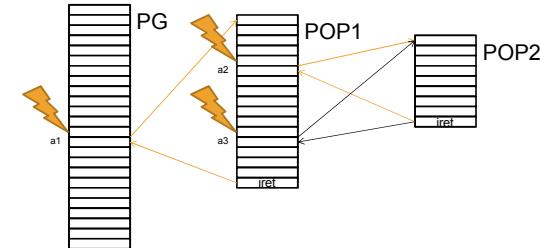
(C) IISI d.KIK PCz 2013

Systemy operacyjne

22

PRZERWANIE O WYŻSZYM PRIORYTECIE

może przerwać POP o niższym priorytecie



(C) IISI d.KIK PCz 2013

Systemy operacyjne

23

(C) IISI d.KIK PCz 2013

Systemy operacyjne

24

WSTĘPNA OBSŁUGA PRZERWAŃ

W wyniku przerwania może nastąpić zmiana stanu procesu:

- ✗ z wykonywanego na wykonywalny
- ✗ z wykonywalnego na wykonywany
- ✗ z niewykonywalnego na wykonywany albo wykonywalny

DYSPOZYTOR

- ✗ Zadaniem dyspozytora jest przydzielanie procesora centralnego do procesów.
- ✗ Dyspozytor dokonuje przełączeń procesów.
- ✗ Dyspozytor jest nazywany planistą niskiego poziomu.

DYSPOZYTOR

Do dyspozytora przechodzi się w przypadku:

- ✗ przerwania zewnętrznego zmieniającego stan jakiegoś procesu
- ✗ po odwołaniu do systemu wstrzymującym wykonywanie bieżącego procesu
- ✗ po wykryciu błędu w bieżącym procesie.

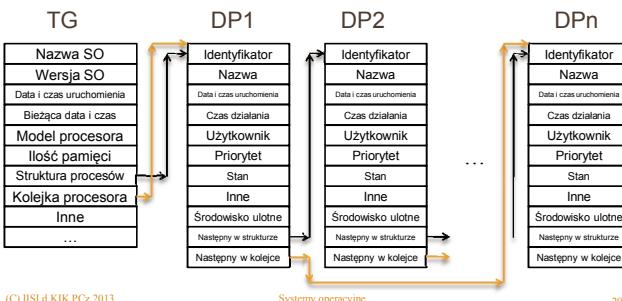
DYSPOZYTOR

Działanie dyspozytora:

- ✗ Jeżeli bieżący proces można kontynuować i ma najwyższy priorytet, to wznow go. Jeżeli nie, to...
 - ✚ Zapamiętaj środowisko ulotne bieżącego procesu.
 - ✚ Odtwórz środowisko ulotne pierwszego procesu do wykonania z kolejki procesora.
 - ✚ Podejmij wykonywanie tego procesu (przekaż mu sterowanie).

KOLEJKA PROCESORA

Jest to lista zawierająca wszystkie procesy wykonywane i wykonywalne, które można w każdej chwili wznowić.



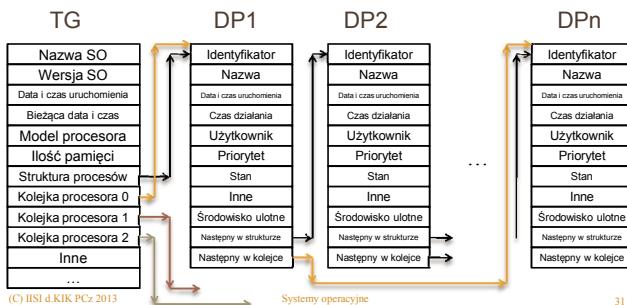
KOLEJKA PROCESORA

Są trzy podstawowe rodzaje kolejek:

- ✗ kolejka priorytetowa - procesy są uporządkowane zgodnie z priorytetami.
- ✗ kolejka rotacyjna - pierwszy proces po wykonaniu przechodzi na koniec kolejki.
- ✗ kolejka wielopoziomowa ze sprzężeniem zwrotnym:
 - ✚ procesy o najwyższym priorytecie są często wybierane, ale na krótki czas,
 - ✚ procesy o najniższym priorytecie są rzadko wybierane, ale na długi czas,
 - ✚ priorytet procesu wzrasta, gdy proces nie wykorzystuje przydzielonego mu czasu,
 - ✚ priorytet procesu maleje, gdy procesowi brakuje czasu.

KOLEJKA PROCESORA

Kolejka wielopoziomowa ze sprzężeniem zwrotnym:



GŁODZENIE PROCESÓW

- ✗ Jest to niemożność wykonania procesu wykonywalnego z powodu długiego oczekiwania na przydział procesora.
- ✗ Dotyczy procesów o niskim priorytecie w mocno obciążonym systemie komputerowym z powodu stałego napływu procesów o wyższych priorytetach.
- ✗ Rozwiązaniem jest podwyższanie priorytetów procesów, które długo oczekują na wykonanie.

(C) IISI d.KIK PCz 2013

Systemy operacyjne

33

IMPLEMENTACJA OPERACJI CZEKAJ I SYGNALIZUJ

- ✗ Operacje czekaj i sygnalizuj działają na semaforach. Semafor jest nieujemną liczbą całkowitą.
- ✗ Operacja czekaj(s):
 - + Zmniejsza wartość semafora o 1, jeżeli jest to możliwe ($s > 0$).
 - + Jest operacją niepodzielną.
 - + Może spowodować wstrzymanie procesu, gdy semafor = 0 do czasu, aż inny proces wykona operację sygnalizuj na tym semaforze.
- ✗ Operacja sygnalizuj(s):
 - + Zwiększa wartość semafora o 1.
 - + Jest operacją niepodzielną.
 - + Może wznowić wykonywanie procesu.

(C) IISI d.KIK PCz 2013

Systemy operacyjne

35

DYSPOZYTOR

Zadaniem dyspozytora jest spowodowanie wykonywania pierwszego procesu z kolejki, który nie jest wykonywany przez inny procesor.

Aby proces stał się wykonywalny należy:

- ✗ zmienić stan w deskryptorze procesu
- ✗ dołączyć deskryptor procesu do kolejki procesora na pozycji wskazywanej przez priorytet procesu.

(C) IISI d.KIK PCz 2013

Systemy operacyjne

32

PODWAŻSZANIE PRIORYTETÓW PROCESÓW INTERAKCYJNYCH

- ✗ Przyznawany procesowi początkowy priorytet (podstawowy) jest traktowany jako punkt startowy.
- ✗ Każdy proces, który podejmie jakieś działanie, otrzymuje tymczasowo wyższy priorytet.
- ✗ Za każdym razem, gdy dyspozytor będzie wybierać dany proces, priorytet tego procesu będzie zmniejszany aż do uzyskania priorytetu podstawowego.
- ✗ Podwyższanie priorytetów jest wykorzystywane do osiągnięcia dobrej pracy interakcyjnej kosztem tych procesów, które wykonują mało pracy interakcyjnej lub w ogóle nie są interakcyjne.

(C) IISI d.KIK PCz 2013

Systemy operacyjne

34

BUDOWA STRUKTURY SEMAFORA

Identyfikator semafora
Nazwa semafora
Wartość
Wartość maksymalna
Wskaznik do kolejki semafora
Organizacja kolejki
Bit użycia

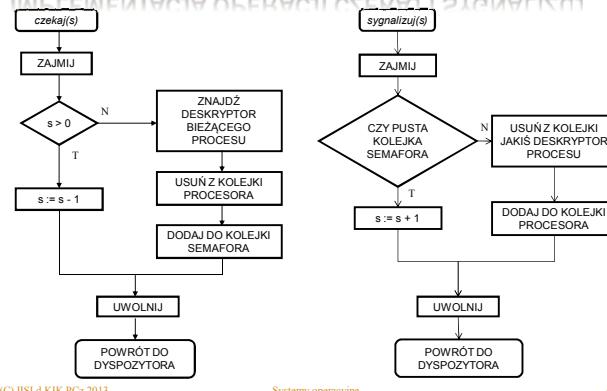
- ✗ Semafony mogą być nazwane lub nie.
- ✗ Kolejka semafora to lista procesów niewykonywalnych, które wykonały operację czekaj na semaforze = 0.
- ✗ Organizacja kolejki zawiera opis organizacji kolejki lub wskaznik do programu dołączającego procesy do kolejki i usuwające je z niej.
- ✗ Bit użycia mówi, czy w danej chwili są wykonywane operacje na semaforze (jest w użyciu). Służy do zapewnienia niepodzielności operacji czekaj i sygnalizuj.

(C) IISI d.KIK PCz 2013

Systemy operacyjne

36

IMPLEMENTACJA OPERACJI CZEKAJ I SYGNALIZUJ

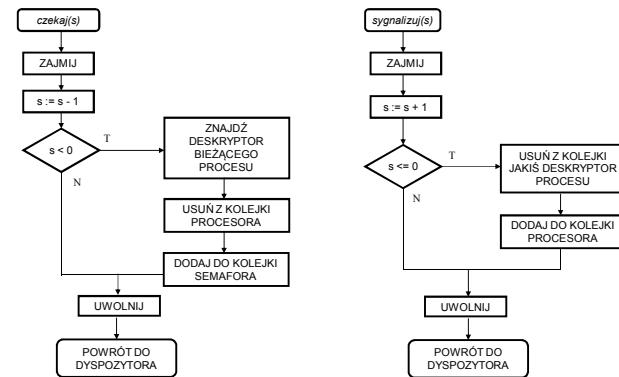


(C) IISI d.KIK PCz 2013

Systemy operacyjne

37

IMPLEMENTACJA OPERACJI CZEKAJ I SYGNALIZUJ



(C) IISI d.KIK PCz 2013

Systemy operacyjne

38

NIEPODZIELNOŚĆ OPERACJI

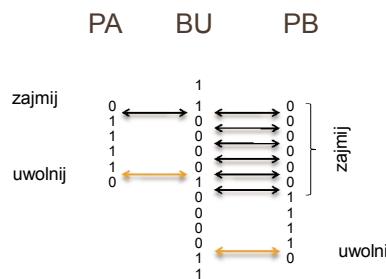
- ✖ W dowolnej chwili tylko jeden proces może wykonywać operacje czekaj i sygnalizuj na tym samym semaforze.
- ✖ Mechanizm ten jest realizowany przez operacje zajmij i uwolnij.

(C) IISI d.KIK PCz 2013

Systemy operacyjne

39

OPERACJE ZAJMIJ I UWOLNIJ



(C) IISI d.KIK PCz 2013

Systemy operacyjne

41

OPERACJE ZAJMIJ I UWOLNIJ

- ✖ Operacja zajmij:
 - + Pewne miejsce w pamięci jest przeznaczane na sygnalizator (bit użycia).
 - + Sprawdza wartość sygnalizatora i ustawia ją na zero (rozkaz „sprawdź i ustaw” (BTR)).
 - + Jeśli wartość sygnalizatora była różna od zera, proces może być wykonywany; w przeciwnym wypadku rozkaz „sprawdź i ustaw” jest wykonywany w pętli aż do zmiany wartości sygnalizatora przez inny proces.
 - + W innej metodzie występuje zamiana wartości sygnalizatora z komórką pamięci, w której wcześniej umieszczono zero (rozkaz swap (XCHG)).
- ✖ Operacja uwolnij:
 - + Nadaje sygnalizatorowi wartość różną od zera.

(C) IISI d.KIK PCz 2013

Systemy operacyjne

40

AKTYWNE CZEKANIE

- ✖ Jest wykonywane wewnątrz operacji zajmij.
- ✖ Jest pętlą, w której podejmowane są próby wykonania operacji aż do chwili zmiany wartości sygnalizatora.

ZARZĄDZANIE PAMIĘCIĄ OPERACYJNA

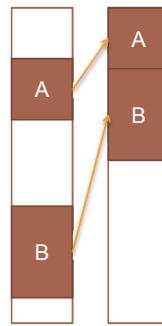
CELE ZARZĄDZANIA PAMIĘCIĄ OPERACYJNA

- ✖ przemieszczanie procesów
- ✖ ochrona zawartości pamięci
- ✖ dostęp do obszarów dzielonych
- ✖ organizacja logiczna
- ✖ organizacja fizyczna pamięci

CELE ZARZĄDZANIA PAMIĘCIĄ OPERACYJNA

Przemieszczanie procesów

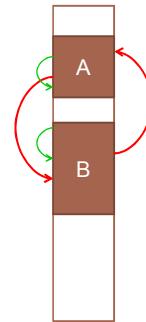
- ✖ procesy mogą być przemieszczane w pamięci w czasie działania, należy zapewnić właściwą zamianę adresów logicznych na fizyczne



CELE ZARZĄDZANIA PAMIĘCIĄ OPERACYJNA

Ochrona zawartości pamięci

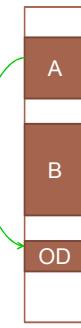
- ✖ sprawdzanie, czy generowane przez proces adresy są mu przydzielone



CELE ZARZĄDZANIA PAMIĘCIĄ OPERACYJNA

Dostęp do obszarów dzielonych

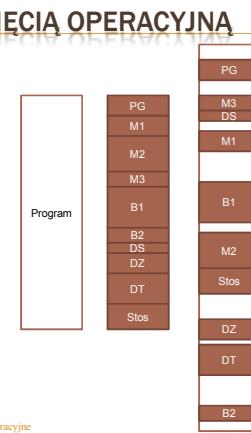
- ✖ część danych lub programów może być wykorzystywana przez kilka procesów, należy zapewnić kontrolę nad dostępem - ochronę przed jednocześnie dostępnym dostępem



CELE ZARZĄDZANIA PAMIĘCIĄ OPERACYJNA

Organizacja logiczna pamięci

- ✖ segmenty programu i danych (kodowane niezależnie, różne poziomy ochrony, dzielenie segmentów między procesy)



CELE ZARZĄDZANIA PAMIĘCIĄ OPERACYJNA

Organizacja fizyczna pamięci

- ✖ pamięć główna
- ✖ pamięć pomocnicza

TWORZENIE ADRESÓW

- ✖ komplikacja
- ✖ ładowanie
- ✖ wykonanie

TWORZENIE ADRESÓW

Kompilacja

- ✖ adres umieszczenia kodu jest znany w trakcie komplikacji



TWORZENIE ADRESÓW

Ładowanie

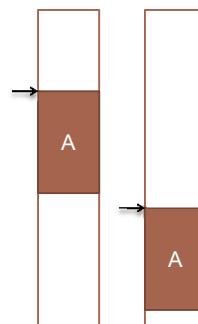
- ✖ adres jest znany w trakcie ładowania



TWORZENIE ADRESÓW

Wykonanie

- ✖ adres może się zmieniać w trakcie wykonywania procesu



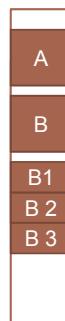
TWORZENIE ADRESÓW

Nakładki



TWORZENIE ADRESÓW

Biblioteki ładowane dynamicznie



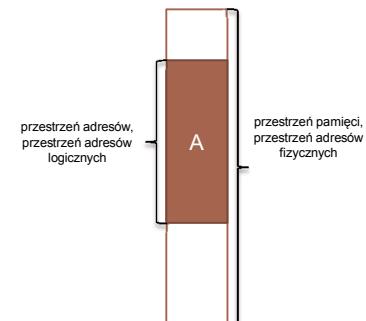
(C) IISI d.KIK PCz 2013

Systemy operacyjne

13

PAMIĘĆ WIRTUALNA

zapewnia odwzorowanie przestrzeni adresów na przestrzeń pamięci



(C) IISI d.KIK PCz 2013

Systemy operacyjne

14

IMPLEMENTACJA PAMIĘCI WIRTUALNEJ

Pamięć wirtualna może być implementowana w formie:

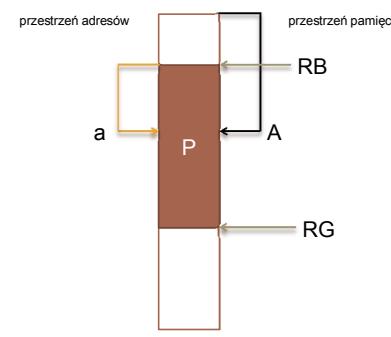
- ✗ Rejestru bazowego i granicznego
- ✗ Stronicowania
- ✗ Segmentacji
- ✗ Segmentacji ze stronicowaniem

(C) IISI d.KIK PCz 2013

Systemy operacyjne

15

REJESTR BAZOWY I GRANICZNY



(C) IISI d.KIK PCz 2013

Systemy operacyjne

16

REJESTR BAZOWY I GRANICZNY

- ✗ $A = f(a) = RB + a$
- ✗ Gdy proces ładowany jest do pamięci, wtedy najmniejszy adres zajęty przez proces jest umieszczany w rejestrze bazowym i wszystkie adresy w programie traktowane są jako adresy **względne** w odniesieniu do adresu bazowego.
- ✗ Odwzorowanie polega zatem na dodawaniu adresu użytego przez proces do adresu bazowego.
- ✗ Przeszczepianie można uzyskać przez przepisanie bloku pamięci i wpisanie nowego adresu do rejestru bazowego.

(C) IISI d.KIK PCz 2013

Systemy operacyjne

17

REJESTR BAZOWY I GRANICZNY

Ochronę pamięci zapewnia rejestr graniczny.

RG – koniec przydzielonego bloku

RG – długość przydzielonego bloku

$$\begin{aligned} A &= f(a) = RB + a \\ A &> RG \end{aligned}$$

$$\begin{aligned} A &= f(a) = RB + a \\ a &> RG \end{aligned}$$

(C) IISI d.KIK PCz 2013

Systemy operacyjne

18

FRAGMENTACJA

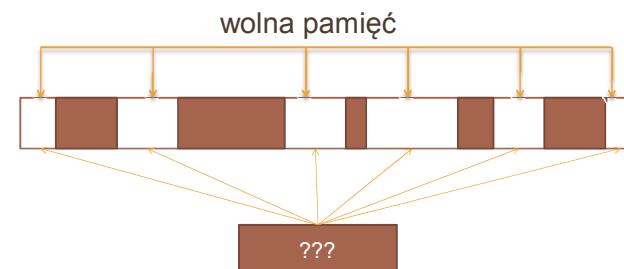
- ✖ W miarę ładowania i usuwania procesów z pamięci operacyjnej pamięć ulega „rozdrobnieniu”. Zjawisko to nazywane jest to **fragmentacją zewnętrzną**.
- ✖ W pamięci powstają „dziury”, po zwolnionych blokach, które nie zostały w pełni zapełnione przez bloki innych procesów.

(C) IISI d.KIK PCz 2013

Systemy operacyjne

19

FRAGMENTACJA ZEWNĘTRZNA



(C) IISI d.KIK PCz 2013

Systemy operacyjne

20

FRAGMENTACJA ZEWNĘTRZNA - UPAKOWANIE

- ✖ Upakowanie pamięci jest stosowane, gdy pojawia znaczna fragmentacja zewnętrzna pamięci.
- ✖ Bloki pamięci przenoszone są tak, aby pozostało w niej najmniej „dziur” lub nie było ich wcale.

przed



po



(C) IISI d.KIK PCz 2013

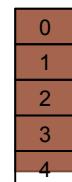
Systemy operacyjne

21

STRONICOWANIE

Przestrzeń adresów jest dzielona na równe strony (2^n ; najczęściej 4 KB)

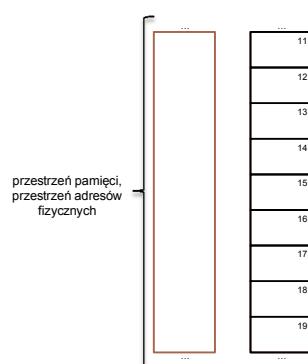
przestrzeń adresów, przestrzeń adresów logicznych



22

STRONICOWANIE

Przestrzeń pamięci dzielona jest na **ramki stron**.



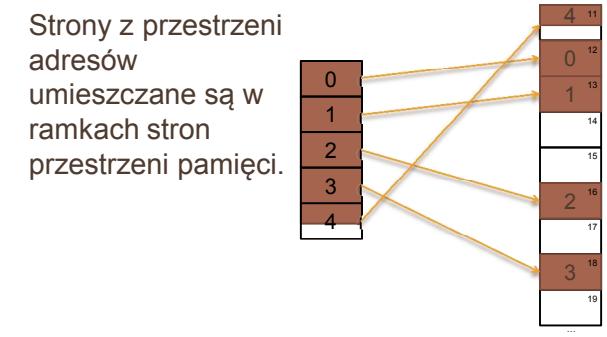
(C) IISI d.KIK PCz 2013

Systemy operacyjne

23

STRONICOWANIE

Strony z przestrzeni adresów umieszczane są w ramkach stron przestrzeni pamięci.



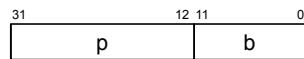
(C) IISI d.KIK PCz 2013

Systemy operacyjne

24

STRONICOWANIE

Dla adresu 32 bitowego i 4 KB strony, adres w przestrzeni adresów (logiczny) można przedstawić:



gdzie:

p – numer strony

b – przesunięcie na stronie

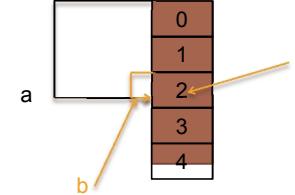
(C) IISI d.KIK PCz 2013

Systemy operacyjne

25

STRONICOWANIE

Graficznie adres można przedstawić:



(C) IISI d.KIK PCz 2013

Systemy operacyjne

26

STRONICOWANIE

Adres w przestrzeni pamięci można wyznaczyć:

$$A = f(a) = f(p, b) = p' + b$$

p' – numer ramki zawierającej stronę p

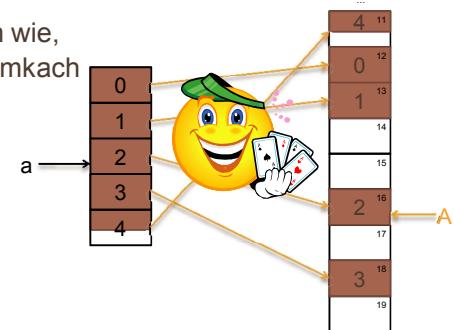
(C) IISI d.KIK PCz 2013

Systemy operacyjne

27

STRONICOWANIE

Pytanie:
skąd system wie,
w których ramkach
są strony?



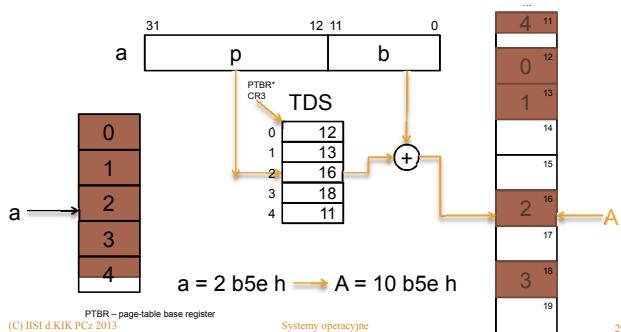
(C) IISI d.KIK PCz 2013

Systemy operacyjne

28

STRONICOWANIE

Tablica deskryptorów stron



PTBR – page-table base register

Systemy operacyjne

29

DESKRYPTOR STRONY

Adres ramki strony	Aval	G	P	A	P	C	W	U	R	I	W	P
--------------------	------	---	---	---	---	---	---	---	---	---	---	---

- ✖ Available for system programmer's use – dostępne dla programistów systemowych
- ✖ Global Page – strona globalna – nie jest usuwana z TLB przy przełączaniu procesów
- ✖ Page Table Attribute Index – PAT pozwala przyporządkować typ pamięci do strony
- ✖ Dirty – strona zmieniona
- ✖ Accessed – strona użyta
- ✖ Cache Disabled – wyłączenie zapamiętywania strony w pamięci podręcznej
- ✖ Write-Through – jednoczesne zapisywanie strony w pamięci podręcznej i głównej
- ✖ User/Supervisor – poziom ochrony użytkowy/systemowy
- ✖ Read/Write – tylko odczyt/zezwolenie na zapis
- ✖ Present – strona obecna

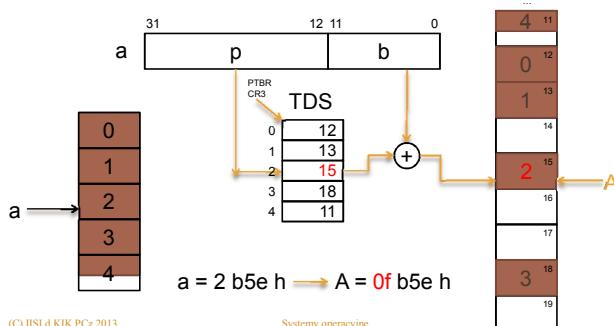
(C) IISI d.KIK PCz 2013

Systemy operacyjne

30

STRONICOWANIE

Przesunięcie strony



(C) IISI d.KIK PCz 2013

Systemy operacyjne

31

STRONICOWANIE

Pamięć wirtualna widziana przez system składa się z:

- ✖ pamięci operacyjnej
- ✖ pamięci pomocniczej (obszaru wymiany) w postaci:
 - + pliku wymiany
 - + partycji wymiany



(C) IISI d.KIK PCz 2013

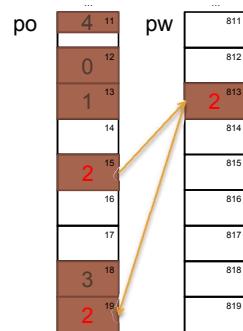
Systemy operacyjne

32

STRONICOWANIE

Korzystanie z pamięci wirtualnej:

- ✖ odesłanie strony z pamięci operacyjnej do pomocniczej
- ✖ i pobranie jej z powrotem z pamięci pomocniczej do operacyjnej w inne miejsce



(C) IISI d.KIK PCz 2013

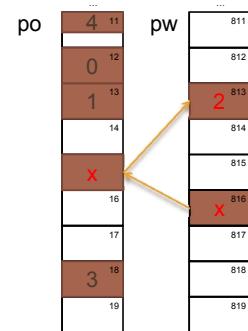
Systemy operacyjne

33

STRONICOWANIE

Korzystanie z pamięci wirtualnej - wymiana:

- ✖ odesłanie strony z pamięci operacyjnej do pomocniczej
- ✖ pobranie innej z pamięci pomocniczej do operacyjnej w jej miejscu



(C) IISI d.KIK PCz 2013

Systemy operacyjne

34

STRONICOWANIE

Zadania:

- ✖ Wykonywanie operacji odwzorowywania adresów, czyli określania, do której ramki odnosi się adres generowany przez proces.
- ✖ Przesyłanie stron z pamięci pomocniczej do pamięci głównej oraz odsyłanie nieużywanych już stron z powrotem do pamięci pomocniczej.

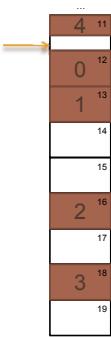
(C) IISI d.KIK PCz 2013

Systemy operacyjne

35

STRONICOWANIE

- ✖ W przypadku stronowania nie ma zjawiska fragmentacji zewnętrznej.
- ✖ Występuje jednak fragmentacja wewnętrzna.
- ✖ Średnio traci się pół ramki na jeden blok pamięci.



(C) IISI d.KIK PCz 2013

Systemy operacyjne

36

STRONICOWANIE

Efektywny czas dostępu do pamięci

$$EAT = 2 * MAT$$

- ✗ MAT – memory access time (np. 100 ns)
- ✗ EAT – effective access time (200 ns)

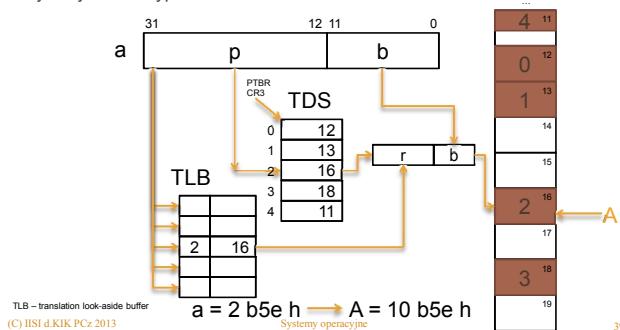
STRONICOWANIE

Pamięć asocjacyjna (skojarzeniowa, TLB-translation look-aside buffer) to:

- ✗ bardzo szybka pamięć wbudowana w procesor
- ✗ adresowana zawartością
- ✗ służąca do przechowywania ostatnio używanych deskryptorów stron (lub segmentów)

STRONICOWANIE

Zastosowanie pamięci asocjacyjnej do przechowywania ostatnio używanych deskryptorów stron



STRONICOWANIE

Efektywny czas dostępu do pamięci z użyciem TLB

$$EAT = HR * (MAT+TLBAT) + (1-HR)*(2* MAT+TLBAT)$$

- ✗ TLBAT – TLB access time (np. 20 ns)
- ✗ HR – hit ratio – współczynnik trafień (np. 80%, 98%)
- ✗ $EAT_{80} = 0,8*120 + 0,2*220 = 140\text{ns}$
- ✗ $EAT_{98} = 0,98*120 + 0,02*220 = 122\text{ns}$

PAMIĘĆ PODRĘCZNA

- ✗ Nazywana również notatnikową lub cache
- ✗ Przechowuje ostatnio używane fragmenty pamięci operacyjnej (kilkadesiąt bajtów) i ich adresy
- ✗ Procesor nie musi odwoływać się do pamięci operacyjnej, jeśli właściwy fragment jest w pamięci notatnikowej
- ✗ Bardzo szybka
- ✗ Najczęściej wbudowana do wnętrza procesora

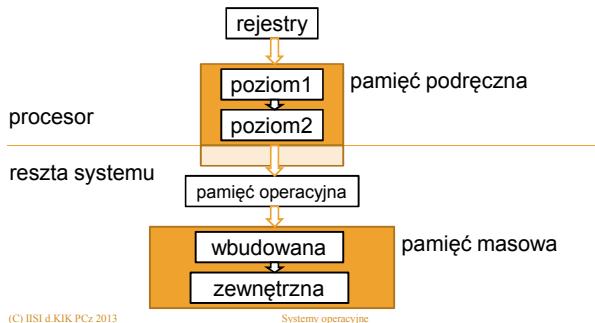
PAMIĘĆ PODRĘCZNA

Może składać się z kilku poziomów:

- ✗ poziom 1 – do kilkudziesięciu kilobajtów
 - ✗ może występować w architekturze Harvard
- ✗ poziom 2 – od kilkudziesięciu kilobajtów do kilkunastu megabajtów
- ✗ ...

PAMIĘĆ PODREczNA

W strukturze pamięci:

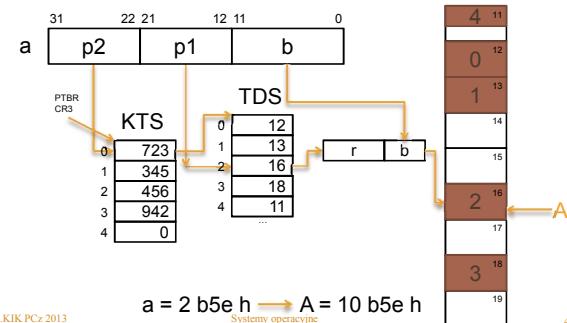


(C) IISI d.KIK PCz 2013

43

STRONICOWANIE

Stronicowanie dwupoziomowe



(C) IISI d.KIK PCz 2013

44

STRONICOWANIE

Efektywny czas dostępu do pamięci

$$EAT = 3 * MAT$$

Efektywny czas dostępu do pamięci z użyciem TLB

$$EAT = HR * (MAT+TLBAT) + (1-HR)*(3* MAT+TLBAT)$$

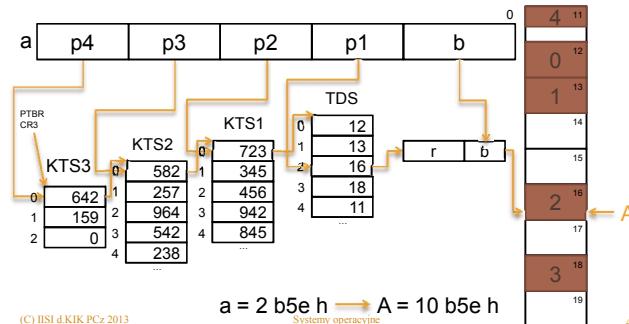
(C) IISI d.KIK PCz 2013

Systemy operacyjne

45

STRONICOWANIE

Stronicowanie wielopoziomowe



(C) IISI d.KIK PCz 2013

46

STRONICOWANIE W PROCESORACH INTEL

31	22 21	12 11	0
p2(10)	p1(10)	b(12)	
31	22 21	0	
p2(10)		b(22)	
31	2120	12 11	0
p2(9)	p1(9)	b(12)	20/24+12
31	2120	12 11	0
p2(9)		b(21)	11/15+21
31	22 21	0	
p2(10)		b(22)	10/14+22
47	3938	3029	2120
p4(9)	p3(9)	p2(9)	p1(9)
47	3938	3029	2120
p4(9)	p3(9)	p2(9)	b(21)

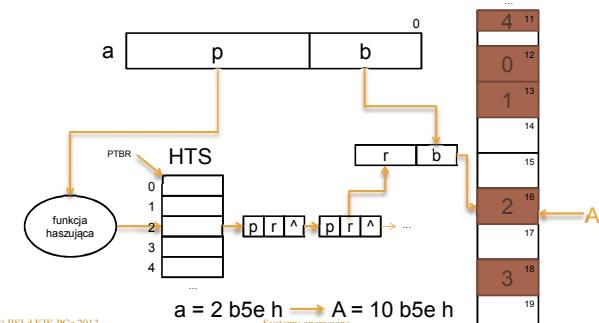
(C) IISI d.KIK PCz 2013

Systemy operacyjne

47

STRONICOWANIE

Haszowana tablica stron – najczęściej dla systemów 64-bitowych. Z powiązanej listy występującej w tablicy wybiera się element dopasowany do strony p.



(C) IISI d.KIK PCz 2013

48

STRONICOWANIE

- ✗ Odmiową haszowaną tablicy stron jest **tablica stron grupowanych**.
- ✗ Każdy wpis odnosi się do np. kilkunastu stron.

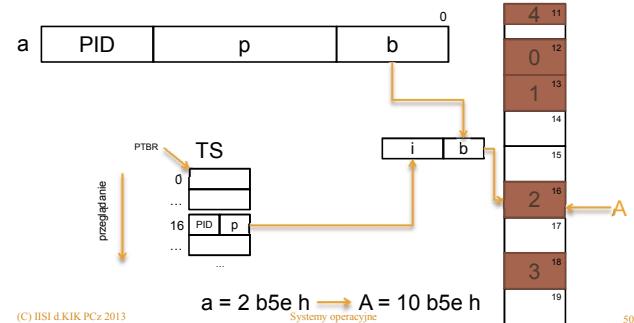
(C) IISI d.KIK PCz 2013

Systemy operacyjne

49

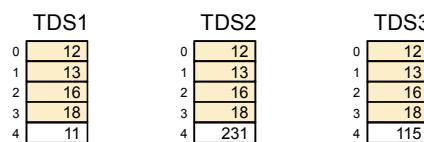
STRONICOWANIE

Odwrócona tablica stron – jedna tablica dla całego systemu – UltraSARC, PowerPC. Często jest haszowana.



STRONICOWANIE

Strony dzielone – część stron może być dzielona przez kilka procesów np. strony zawierające kod (wznawialny - czysty) programu lub stałe.



(C) IISI d.KIK PCz 2013

Systemy operacyjne

51

STRONICOWANIE

Stronicowanie na żądanie – do pamięci sprowadzanie są tylko niektóre strony procesu (*leniwa wymiana*), pozostałe są pobierane tylko wówczas, gdy są potrzebne.

Przebieg:

- ✗ błęd braku strony
- ✗ sprowadzenie brakującej strony do wolnej ramki
- ✗ uaktualnienie tablicy stron
- ✗ wznowienie procesu

(C) IISI d.KIK PCz 2013

Systemy operacyjne

52

STRONICOWANIE

Sprawność stronicowania:

$$EAT = (1-p) * MAT + p * PFST$$

- ✗ PFST – page fault service time – czas obsługi błędu braku strony
- ✗ p – prawdopodobieństwo wystąpienia błędu braku strony

(C) IISI d.KIK PCz 2013

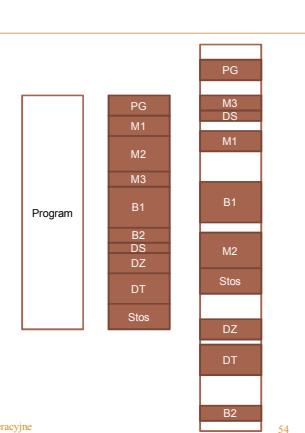
Systemy operacyjne

53

SEGMENTACJA

Segmentacja polega na podziale pamięci przydzielonej procesowi na niezależne fragmenty zwane **segmentami**.

Odzwierciedla ona organizację logiczną pamięci.



(C) IISI d.KIK PCz 2013

Systemy operacyjne

54

SEGMENTACJA

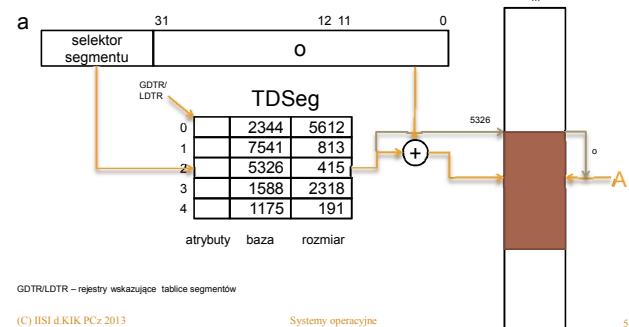
Adres w przestrzeni pamięci wyznacza się:

$$A = f(a) = B + o$$

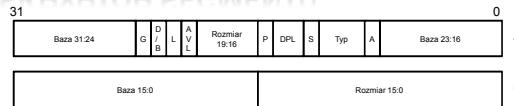
B – adres bazowy segmentu
o – przesunięcie w segmencie

SEGMENTACJA

Adresowanie w segmentacji



DESKRYPTOR SEGMENTU



- ✗ G — Granularity – ziarnistość – rozmiar liczony w bajtach (0) lub 4KB (1)
- ✗ D/B — Default operation size – rozmiar operacji (0 = 16-bit; 1 = 32-bit)
- ✗ L — 64-bit code segment – segment zawiera kod 64-bitowy (IA-32e mode only)
- ✗ AVL — Available for use by system software – dostępne dla programów systemowych
- ✗ DPL — Descriptor privilege level – poziom ochrony
- ✗ P — Segment present – segment obecny w pamięci
- ✗ S — Descriptor type - rodzaj (0 = systemowy; 1 = kod/dane)
- ✗ TYPE — Segment type - typ segmentu (dane, kod, odczyt, zapis, wykonywanie)
- ✗ A — Accessed - segment użyty

SEGMENTACJA

Mogą występować dwie tablice deskryptorów segmentów:

✗ globalna

+ dla systemu operacyjnego

✗ lokalna

+ dla każdego procesu

+ informacja o niej znajduje się w tablicy globalnej

SELEKTOR SEGMENTU



- ✗ RPL — Requestor's privilege level – poziom ochrony procesu wywołującego
- ✗ TI — Table indicator – wskaźnik rodzaju tablicy
- ✗ CPL – bieżący poziom ochrony
- ✗ EPL – efektywny poziom ochrony EPL=max(CPL,RPL)
- ✗ Dla danych musi być $EPL \leq DPL$
 - + można odwoływać się do danych o niższym/równym poziomie ochrony
- ✗ Dla kodu $CPL \geq DPL$
 - + można wywoływać programy o wyższym/równym poziomie ochrony

SEGMENTACJA

Segmenty mogą występować w kilku tablicach deskryptorów segmentów, stają się wówczas **segmentami dzielonymi**.

TDS1		TDS2		TDS3	
0	1256	0	8172	0	1670
1	5279	1	9731	1	4724
2	16	2	16	2	16
3	18	3	18	3	18
4	11	4	231	4	115
...		

SEGMENTACJA A STRONICOWANIE

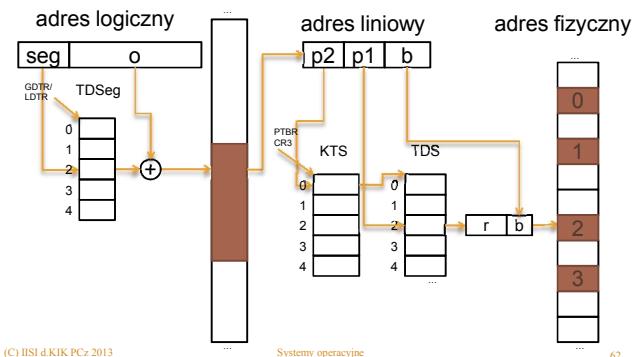
- ✗ Celem segmentacji jest logiczny podział pamięci, natomiast stronicowania fizyczny.
- ✗ Strony mają ustalony rozmiar, segmenty mogą być dowolne.
- ✗ Stronicowanie odbywa się w sposób sprzętowy i przekroczenie zakresu strony powoduje przejście do następnej, w segmentacji podział adresu jest logiczny i nie powoduje przejścia do następnego segmentu.
- ✗ Segmentacja zapewnia poziomy ochrony kodu i danych.

(C) IISI d.KIK PCz 2013

Systemy operacyjne

61

SEGMENTACJA ZE STONICOWANIEM



(C) IISI d.KIK PCz 2013

Systemy operacyjne

62

SEGMENTACJA ZE STONICOWANIEM

- ✗ Segmentacja ze stronicowaniem to połączenie cech dwóch metod implementacji pamięci wirtualnej (segmentacji i stronicowania).
- ✗ Polega ona na stronicowaniu segmentów. Pozwala to wyeliminować fragmentację zewnętrzną przy uproszczeniu przydziału:
 - + każda pusta ramka może być użyta na potrzebną stronę.
- ✗ W takiej organizacji pamięci każdy segment może posiadać własną tablicę stron. Przy czym nie musi ona być pełna, a zawiera tyle stron ile jest potrzebne segmentowi.

(C) IISI d.KIK PCz 2013

Systemy operacyjne

63

STRATEGIE PRZYDZIAŁU PAMIĘCI

- Strategie przydziału pamięci dzieli się na:
- ✗ strategie rozmieszczenia
 - ✗ strategie wymiany
 - ✗ strategie pobierania

(C) IISI d.KIK PCz 2013

Systemy operacyjne

64

STRATEGIE PRZYDZIAŁU PAMIĘCI

- ✗ **Przydział ciągły** – podzielenie pamięci na obszary. Jeden obszar może zajmować dokładnie jeden proces.
- ✗ Stopień wieloprogramowości ograniczony przez liczbę obszarów.
- ✗ Przydział ciągły nie jest dobrym rozwiązaniem ze względu na fragmentację wynikającą z pozostałych po procesach dziur.

(C) IISI d.KIK PCz 2013

Systemy operacyjne

65

STRATEGIE PRZYDZIAŁU PAMIĘCI

Strategie rozmieszczenia:

- ✗ służą do określania **miejsca** w pamięci głównej, do którego mają być załadowane dane lub kod, to znaczy do wybierania podzbioru niezajętych obszarów

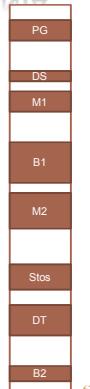
(C) IISI d.KIK PCz 2013

Systemy operacyjne

66

ROZMIESZCZENIE BEZ STRONICOWANIA

- ✖ System prowadzi ewidencję dziur, wiadomo ile jest dziur i jakie są duże
- ✖ Wyróżniamy następujące strategie rozmieszczenia:
 - + najlepsze dopasowanie
 - + najgorsze dopasowanie
 - + pierwsze dopasowanie
 - + algorytm bliźniaków



67

ROZMIESZCZENIE BEZ STRONICOWANIA

Najlepsze dopasowanie

- ✖ dziury są uporządkowane rosnąco pod względem rozmiaru
- ✖ system szuka bloku najmniejszego spełniającego warunek rozmiaru

ROZMIESZCZENIE BEZ STRONICOWANIA

Najgorsze dopasowanie

- ✖ dziury są uporządkowane malejąco pod względem rozmiaru
- ✖ system umieszcza proces w pierwszej dziurze na liście

69

ROZMIESZCZENIE BEZ STRONICOWANIA

Pierwsze dopasowanie

- ✖ dziury są uporządkowane rosnąco pod względem ich adresów bazowych
- ✖ system przydziela pierwszą z możliwych dziur

ROZMIESZCZENIE BEZ STRONICOWANIA

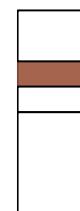
Algorytm bliźniaków

- ✖ Rozmiar dziur jest zawsze $= 2^n$
- ✖ Gdy potrzebna ilość pamięci jest mniejsza niż połowa dziury, system dzieli wybraną dziurę na dwie równe części do momentu aż dziura będzie odpowiednia. Blok umieszczany w powstałej dziurze.
- ✖ Jeśli nie ma wystarczająco dużej dziury system łączy dwie takie same sąsiadujące dziury w jedną dwa razy większą.

71

ROZMIESZCZENIE BEZ STRONICOWANIA

Algorytm bliźniaków



ROZMIESZCZENIE ZE STRONICOWANIEM

- ✖ Nie ma potrzeby przechowywania listy dziur
- ✖ Występuje lista wolnych stron
- ✖ Nie powoduje fragmentacji zewnętrznej
- ✖ Występuje fragmentacja wewnętrzna

(C) IISI d.KIK PCz 2013

Systemy operacyjne

73

STRATEGIE PRZYDZIAŁU PAMIĘCI

Strategie wymiany:

- ✖ służą do określenia bloków, które można usunąć z pamięci operacyjnej, aby powstały w niej wolne obszary

(C) IISI d.KIK PCz 2013

Systemy operacyjne

74

WYMIANA ZE STRONICOWANIEM

Proces żąda dostępu do strony w obszarze wymiany. W przypadku braku wolnych ramek na dysk jest odsyłana strona:

- ✖ najdawniej używana
- ✖ najmniej używana
- ✖ najdawniej załadowana

(C) IISI d.KIK PCz 2013

Systemy operacyjne

75

WYMIANA ZE STRONICOWANIEM

Wymiana strony **najdawniej używanej**:

- ✖ w deskryptorze strony musi być zapisany czas odwołania do strony
- ✖ stosowanie tego algorytmu jest obciążone kosztami zapisywania czasu odwołań do wszystkich stron

(C) IISI d.KIK PCz 2013

Systemy operacyjne

76

WYMIANA ZE STRONICOWANIEM

Wymiana **najmniej używanej** strony

- ✖ w tym algorytmie należy odnotowywaćczęstość użycia strony
- ✖ wadą tej metody jest fakt, że ostatnio załadowana strona może być nieroźważnie wymieniona
- ✖ można zabronić wymiany stron załadowanych podczas określonego przedziału czasu

(C) IISI d.KIK PCz 2013

Systemy operacyjne

77

WYMIANA ZE STRONICOWANIEM

Wymiana **najdawniej załadowanej** strony:

- ✖ algorytm ten wymaga zapisu ciągu stron załadowanych do pamięci, co pozwala wskazać najdawniej załadowane strony
- ✖ pomija się w nim fakt, że najwięcej odniesień może być do strony, która najdłużej znajduje się w pamięci

(C) IISI d.KIK PCz 2013

Systemy operacyjne

78

WYMIANA BEZ STRONICOWANIA

Wymieniane segmenty posiadają różny rozmiar, zależy od tego sposób wymiany

- ✖ jeśli do pamięci ma być załadowany mały segment wystarczy odesłać tylko mały segment
- ✖ jeśli segment jest duży, to trzeba odesłać duży segment lub kilka mniejszych
- ✖ może wystąpić konieczność upakowania pamięci

WYMIANA BEZ STRONICOWANIA

- ✖ Najprostszy algorytm polega na zastąpieniu jednego segmentu oraz sąsiedniej dziury, których łączny obszar wystarczy do pomieszczenia nowego segmentu
- ✖ Gdy takich segmentów jest kilka można zastosować algorytm najdawniej załadowanych, bądź najmniej używanych segmentów

STRATEGIE PRZYDZIAŁU PAMIĘCI

Strategie pobierania:

- ✖ służą do określania momentu, w którym bloki mają być załadowane do pamięci operacyjnej z obszaru wymiany

STRATEGIE POBIERANIA

Strategie pobierania z pamięci pomocniczej:

- ✖ pobieranie na żądanie
- ✖ pobieranie przewidujące

STRATEGIE POBIERANIA

Pobieranie na żądanie:

- ✖ wystąpienie błędu braku bloku (segmentu lub strony) generuje żądanie pobrania bloku, a strategie rozmieszczania służą do przydziału pamięci dla nowego bloku
- ✖ w systemach bez stronicowania przesyła się bloki najczęściej na żądanie

STRATEGIE POBIERANIA

Pobieranie przewidujące:

- ✖ system może pobierać nowe bloki w oparciu o:
 - + znajomość konstrukcji programu
 - + wnioskowanie z dotychczasowego działania procesu

ZASADA LOKALNOŚCI

- ✖ Wiele programów wykazuje cechę zwaną **działaniem w kontekście**.
- ✖ Oznacza to, że w dowolnie małych przedziałach czasu program działa wewnątrz konkretnego modułu logicznego.
- ✖ Wykonuje on rozkazy należące do jednej procedury lub pobiera dane z jednego obszaru.



(C) IISI d.KIK PCz 2013

Systemy operacyjne

85

(C) IISI d.KIK PCz 2013

Systemy operacyjne

86

ZASADA LOKALNOŚCI

- ✖ Odniesienia do pamięci wydają się grupować wewnątrz małych obszarów przestrzeni adresów
 - + lokalność odniesień jest jeszcze większa w przypadku częstego występowania pętli
- ✖ Do zmiany tych obszarów dochodzi tylko okresowo.
- ✖ **Procesy chwilowo wykonują się lokalnie.**

SZAMOTANIE

- ✖ Gdy proces nie ma wystarczającej ilości ramek, szybko może nastąpić brak strony.
- ✖ Stronę brakującą system sprowadza do pamięci operacyjnej, wymieniając z inną stroną.
- ✖ Może dojść do sytuacji, w której wymieniona strona (już nieobecna w pamięci) jest potrzebna dla aktywnego procesu, następuje ponowne jej sprowadzenie do pamięci.
- ✖ W rezultacie w systemie będą następowały po sobie kolejne braki stron i ich wymiany.
- ✖ Procesy będą wskazywać braki stron, system dokona wymiany stron, po czym sprowadzi je z powrotem.

(C) IISI d.KIK PCz 2013

Systemy operacyjne

87

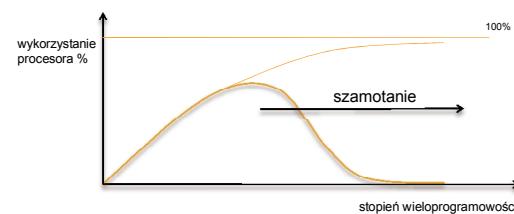
(C) IISI d.KIK PCz 2013

Systemy operacyjne

88

SZAMOTANIE

Szamotanie to duża aktywność stronicowania, system spędza więcej czasu na stronicowaniu (wymianie) niż na wykonaniu procesów.



SZAMOTANIE

- ✖ Efekt szamotania można ograniczyć za pomocą **lokalnego algorytmu zastępowania**
 - + Przy zastępowaniu lokalnym, gdy jakiś proces zaczyna się szamotać wówczas nie wolno mu zabierać ramek innego procesu i doprowadzać go także do szamotania
- ✖ Aby zapobiec szamotaniu należy dać procesowi tyle ramek, ile potrzebuje.

(C) IISI d.KIK PCz 2013

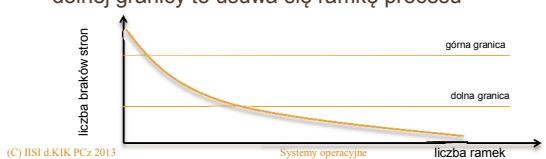
Systemy operacyjne

89

SZAMOTANIE

Metodą mierzenia szamotania jest mierzenie częstości braków stron:

- ✖ ustala się górną i dolną granicę pożdanego poziomu braków stron
- ✖ jeśli proces przekracza górną granicę, to dostaje dodatkową ramkę
- ✖ jeśli częstość występowania braków stron spada poniżej dolnej granicy to usuwa się ramkę procesu



SZAMOTANIE – MODEL STREFOWY

W trakcie wykonywania proces przechodzi z jednej strefy do innej.

- ✗ Strefa to zbiór stron pozostających we wspólnym użytku.
- ✗ Program składa się z wielu stref, które mogą na siebie zachodzić.
- ✗ Procesowi przydziela tyle ramek, aby mógł w nich zmieścić swoją bieżącą strefę.
- ✗ Gdy przydzielone jest mniej ramek, niż wynosi rozmiar bieżącej strefy, wówczas proces zaczyna się szamotać.

MODEL ZBIORU ROBOCZEGO

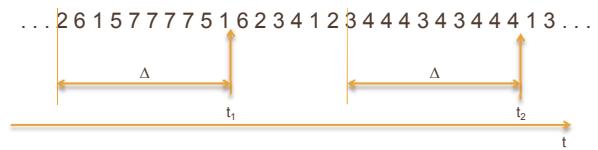
- ✗ Opiera się na założeniu, że program ma charakterystykę strefową.
- ✗ Parametr Δ definiuje okno zbioru roboczego.
- ✗ **Za zbiór roboczy przyjmuje się zbiór stron, do których nastąpiło Δ ostatnich odwołań.**

MODEL ZBIORU ROBOCZEGO

- ✗ **Jeśli zbiór roboczy procesu znajduje się w pamięci, wówczas proces wykonuje się bez szamotania.**
- ✗ Strona aktywnie używana znajduje się w zbiorze roboczym.
- ✗ Gdy strona przestanie być używana, wypada ze zbioru roboczego po Δ jednostkach czasu liczących od ostatniego odwołania do niej.

MODEL ZBIORU ROBOCZEGO

Ślad odniesień:



Dla $\Delta = 10$

odpowiednie zbiory robocze wynoszą:

$$ZR(t_1) = \{1, 2, 5, 6, 7\}$$

$$ZR(t_2) = \{3, 4\}$$

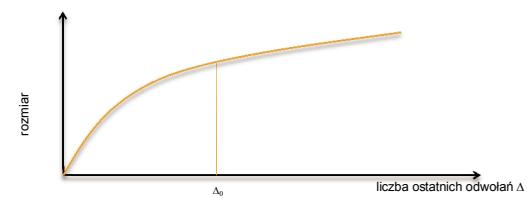
MODEL ZBIORU ROBOCZEGO

Dokładność zbioru roboczego zależy od wyboru parametru Δ :

- ✗ Jeśli jest za mały – nie obejmie całego zbioru
- ✗ Jeśli za duży, to może zachodzić na kilka stref programu
- ✗ W skrajnym przypadku, gdy Δ jest nieskończoność duża, zbiorem roboczym staje się zbiór stron, z którymi proces kontaktował się w trakcie działania.

MODEL ZBIORU ROBOCZEGO

Oczekiwany rozmiar zbioru roboczego.



OBSŁUGA WEJŚCIA-WYJŚCIA

✗ System operacyjny musi komunikować się ze światem zewnętrznym.

✗ Obsługa wejścia-wyjścia jest jednym z najmudniejszych etapów tworzenia systemu.

✗ powodem jest konieczność stosowania ogólnych technik do różnych urządzeń zewnętrznych, o różnych parametrach technicznych i trybach pracy.

RÓŻNICE POMIĘDZY URZĄDZENIAMI WEJŚCIA WYJŚCIA

✗ Zastosowanie

✗ przechowywanie danych, komunikacja, sterowanie

✗ Prędkość

✗ sieć, dysk: 10^7 - 10^{12} , klawiatura 10^0 znaków na sekundę

✗ Jednostki przesyłania

✗ bity, bajty, słowa

✗ Sposób przesyłania

✗ strumień, blok

✗ Reprezentacja danych

✗ różne sposoby kodowania na różnych nośnikach

✗ Operacje uprzywilejowane

✗ druk na drukarce, czytanie z płyty optycznej

RÓŻNICE POMIĘDZY URZĄDZENIAMI WEJŚCIA WYJŚCIA C.D.

✗ Czytanie lub pisanie

✗ klawiatura, drukarka

✗ Dostęp sekwencyjny lub swobodny

✗ taśma, dysk

✗ Synchroniczność lub asynchronouszność

✗ Urządzenie dzielone lub na wyłączność

✗ Złożoność sterowania

✗ drukarka, dysk

✗ Różne przyczyny wystąpienia błędów

✗ brak papieru w drukarce, zła suma kontrolna

URZĄDZENIA WEJŚCIA-WYJŚCIA

najczęściej posiadają rejestyry

✗ stanu

✗ sterowania

✗ danych wejściowych

✗ danych wyjściowych

ADRESOWANIE URZĄDZEŃ WEJŚCIA-WYJŚCIA

Procesor odwołuje się do rejestrów urządzeń poprzez porty z przestrzeni:

✗ we/wy

✗ instrukcje in i out

✗ pamięci

✗ instrukcja mov i wszystkie tryby adresowania

REALIZACJE OPERACJI WEJŚIA-WYJŚCIA

- ✖ Programowe wejście-wyjście - odpytywanie
 - ✖ procesor zleca obsługę i odpytuje urządzenie do jej zakończenia
- ✖ Operacje wejścia-wyjścia sterowanie przerwaniami
 - ✖ kolejne dane są transmitowane w kolejnych przerwaniach
- ✖ Bezpośredni dostęp do pamięci (DMA)
 - ✖ procesor zleca wykonanie transmisji kończącej się przerwaniem

(C) IISI d.KIK PCz 2013

Systemy operacyjne

7

(C) IISI d.KIK PCz 2013

Systemy operacyjne

8

ROZWÓJ OBSŁUGI WEJŚIA-WYJŚCIA

- ✖ Sterowanie bezpośrednio przez procesor
- ✖ Procesor zleca wykonanie operacji kontrolerowi, który bezpośrednio steruje urządzeniem
- ✖ Zastosowanie kontrolera z przerwaniami
- ✖ Zastosowanie kontrolera z obsługą DMA
- ✖ Zastosowanie procesora we/wy, który po otrzymaniu polecenia od procesora głównego wykonuje właściwy podprogram
- ✖ Procesor we/wy staje się oddzielnym komputerem z własną pamięcią

WYMAGANIA STAWIANE SYSTEMOWI

- ✖ Niezależność od kodów znaków
 - ✖ identyczna interpretacja dla wszystkich urządzeń
- ✖ Niezależność od urządzeń wejścia – wyjścia
 - ✖ tego samego typu: np. który napęd, czy drukarka
 - ✖ różnych typów: np. przesłanie na dysk lub drukarkę realizuje prawie identyczny program
- ✖ Uzyskanie jak największej wydajności
 - ✖ ze względu na powstawanie tzw. wąskiego gardła
- ✖ Jednolite traktowanie urządzeń
 - ✖ celem uproszczenie obsługi i zmniejszenie liczby błędów

(C) IISI d.KIK PCz 2013

Systemy operacyjne

9

(C) IISI d.KIK PCz 2013

Systemy operacyjne

10

SYSTEMOWE MECHANIZMY OBSŁUGI WEJŚIA – WYJŚCIA

- ✖ Kodowanie znaków w urządzeniach jest często różne i niezgodne z kodowaniem w systemie.
- ✖ Wewnętrzny kod znaków – kod, przez który wykonuje się tłumaczenie znaków dla poszczególnych urządzeń.
- ✖ Stosuje się tablice przekodowujące, aby znaki wewnętrz systemu były reprezentowane w sposób jednoznaczny.

SYSTEMOWE MECHANIZMY OBSŁUGI WEJŚIA – WYJŚCIA

Urządzenia wirtualne

- ✖ nazywane strumieniami, plikami, zbiorami danych
- ✖ poprzez nie następuje komunikacja z urządzeniami fizycznymi
- ✖ odwołując się do urządzenia najpierw należy uzyskać dostęp do niego od systemu, a po zakończeniu użytkowania należy je zwolnić.

(C) IISI d.KIK PCz 2013

Systemy operacyjne

11

(C) IISI d.KIK PCz 2013

Systemy operacyjne

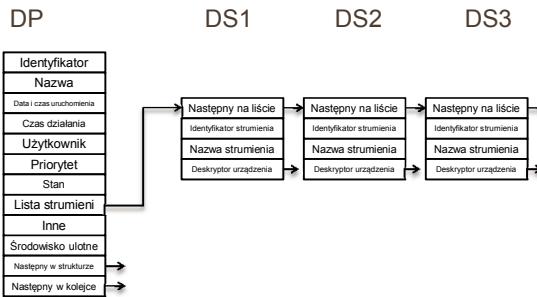
12

SYSTEMOWE MECHANIZMY OBSŁUGI WEJŚIA – WYJŚCIA

- ✖ Programy działają na wirtualnych urządzeniach (strumieniach) zamiast na rzeczywistych.
- ✖ Po przypisaniu urządzenia do strumienia
 - + define output1 tape3

programista kieruje wyjście na strumień, a przesłaniem do urządzenia zajmuje się system operacyjny.
- ✖ Dzięki temu programy obsługi urządzeń mogą być do siebie podobne
- ✖ Różnice są ukryte przez system operacyjny.
- ✖ Informacje o właściwościach urządzenia znajdują się w deskryptorze urządzenia.

LISTA DESKRYPTORÓW STRUMIENI



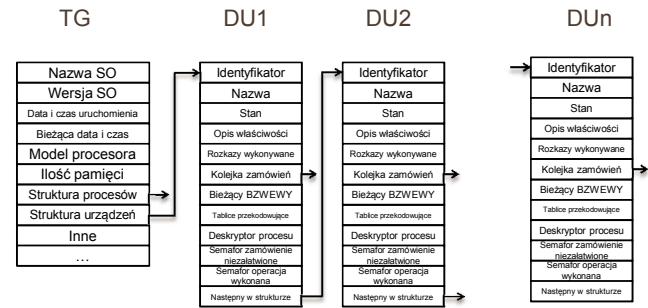
Nazwy strumieni muszą być unikalne.

(C) IISI d.KIK PCz 2013

Systemy operacyjne

13

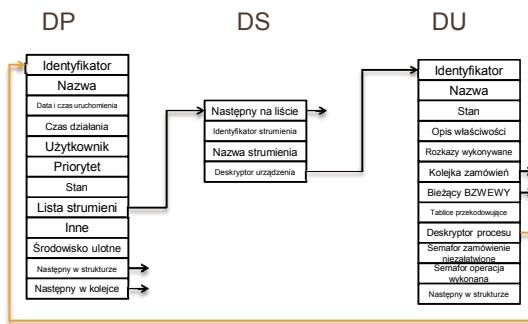
STRUKTURA URZĄDZEŃ I DESKRYPTORY URZĄDZEŃ



Struktura urządzeń wskazuje z tablicy głównej listę deskryptorów urządzeń
(C) IISI d.KIK PCz 2013 Systemy operacyjne

14

POWIĄZANIE DESKRYPTORÓW



(C) IISI d.KIK PCz 2013

Systemy operacyjne

15

PODPROGRAMY OBSŁUGI URZĄDZEŃ

- ✖ Programy te muszą być wielowejściowe, tzn. wiele procesów może się do nich odwoływać.
- ✖ Ich podstawowe zadania to:
 - ✖ tworzenie bloków zamówień wejścia – wyjścia,
 - ✖ informowanie procesu obsługi, że został dla niego przygotowany blok zamówienia.

(C) IISI d.KIK PCz 2013 Systemy operacyjne

16

PLIKI SPECJALNE

- ✖ Pliki specjalne można rozumieć jako wirtualne urządzenia.
- ✖ Wiele funkcji systemowych dla plików jest również potrzebnych w przypadku urządzeń. Po zażądaniu przydziału urządzenia (i uzyskaniu go) można wykonywać na nim operacje czytania, pisania i (być może) zmiany położenia - jak w operacjach na zwykłych plikach.
- ✖ Podobieństwo między urządzeniami wejścia-wyjścia i plikami może być tak duże, że wiele systemów operacyjnych (UNIX, MS-DOS) łączy je w jedną strukturę plików - urządzeń.

(C) IISI d.KIK PCz 2013

Systemy operacyjne

17

PLIKI SPECJALNE

- ✖ W systemach typu UNIX pliki specjalne do obsługi urządzeń znajdują się w katalogu /dev .
- ✖ W systemie MS-DOS można wyróżnić pewne urządzenia, które są traktowane jak pliki.
 - ✖ port równoległy LPT,
 - ✖ port szeregowy COM,
 - ✖ konsola CON.

(C) IISI d.KIK PCz 2013 Systemy operacyjne

18

PROCEDURY WEJŚCIA-WYJŚCIA

Proces użytkownika zgłasza zamówienia do systemu korzystając z funkcji:

OPWEWY(strumień, tryb, rozmiar, lokalizacja, semafor)

gdzie:

- + OPWEWY – nazwa procedury systemowej
- + strumień – numer strumienia we/wy
- + tryb – tryb dostępu (wysłanie danych, wysunięcie papieru)
- + rozmiar – rozmiar danych
- + lokalizacja – źródło lub miejsce przeznaczenia danych
- + semafor – adres semafora, do którego zostanie przesłany sygnał zakończenia operacji

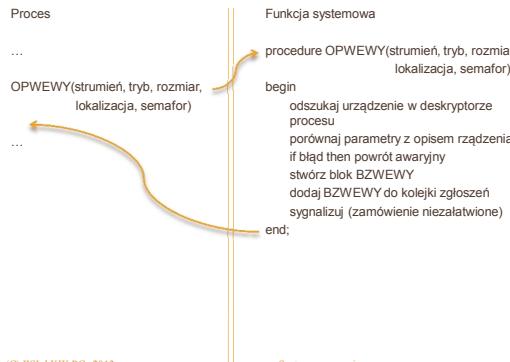
PROCEDURY WEJŚCIA-WYJŚCIA

- ✖ Procedura OPWEWY jest wielowejściowa (może być wywołana przez kilka procesów naraz).

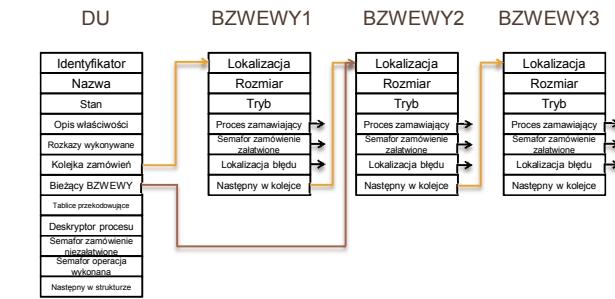
Przebiega ona w następujących krokach:

- ✖ odszukanie urządzenia w deskryptorze procesu
- ✖ sprawdzenie parametrów wywołania procedury z opisem urządzenia
- ✖ zainicjowanie obsługi urządzenia
 - + utworzenie bloku zamówień wejścia-wyjścia (BZWEWY) i dodanie go do kolejki zamówień urządzenia
 - + wysłanie sygnału do procesu obsługi urządzenia informującego o nowym zamówieniu
 - ✖ sygnalizuj (zamówienie niezałatwione)

PROCEDURA WEJŚCIA-WYJŚCIA



DESKRYPTOR URZĄDZENIA I KOLEJKA ZAMÓWIENI



Kolejka zamówień może być dowolnie zorganizowana.

PROCESY OBSŁUGI URZĄDZEŃ

- ✖ Obsługą zamówień umieszczonych w kolejce urządzenia oraz powiadamianiem procesów-nadawców o wykonaniu operacji we/wy zajmuje się proces obsługi urządzeń.
- ✖ Każde urządzenie posiada osobny proces, jednak podobieństwa pozwalają użyć wspólnych programów.
- ✖ Różnice spowodowane są odmiennymi cechami konkretnych urządzeń.

PROCES OBSŁUGI URZĄDZENIA WYJŚCIOWEGO

```

while true do {obsługa wyjścia}
begin
  czekaj (zamówienie niezałatwione)
  wybór BZWEWY z kolejki zamówień
  wydobycie szczegółów zamówienia
  prześlij dane z podanej lokalizacji
  tłumacz znaki
  zainicjowanie operacji we-wy
  czekaj (operacja wykonana)
  if błędny then umieść informację o błędzie
  sygnalizuj (zamówienie załatwione)
  usunięcie BZWEWY
end;
  
```

Dla transmisji poprzez kanał; można zastąpić odpitywaniem!

PODPROGRAM OBSŁUGI PRZERWANIA

Transmisja przez kanał kończy się przerwaniem:

POP:

lokalizacja deskryptora urządzenia
sygnalizuj(operacja wykonana)

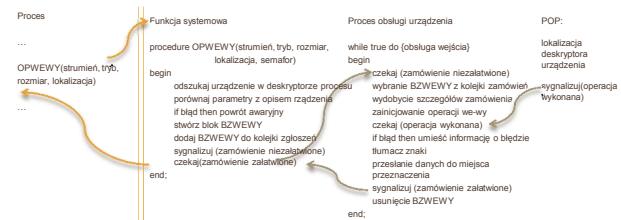
PROCES OBSŁUGI URZĄDZENIA WEJŚCIOWEGO

```
while true do {obsługa wejścia}
begin
    czekaj (zamówienie niezałatwione)
    wybranie BZWEWY z kolejki zamówień
    wydobycie szczegółów zamówienia
    zainicjowanie operacji we-wy
    czekaj (operacja wykonana)
    if błęd then umieść informację o błędzie
    tłumacz znaki
    przesłanie danych do miejsca przeznaczenia
    sygnalizuj (zamówienie załatwione)
    usunięcie BZWEWY
end;
```

CAŁOŚCIOWA PROCEDURA WEJŚCIA-WYJŚCIA



CAŁOŚCIOWA PROCEDURA WEJŚCIA-WYJŚCIA



OBSŁUGA WEJŚCIA-WYJŚCIA

może odbywać się na dwa sposoby:

- ✖ proces zamawiający może wykonywać się równolegle z operacją we/wy
 - + sam musi zadbać o własne wstrzymanie przed odwołaniem się do informacji pochodzącej z wywołanej operacji; musi on wykryć zakończenie operacji we/wy
- ✖ procedura we/wy, będąca częścią systemu operacyjnego, wstrzymuje proces do zakończenia operacji we/wy
 - + semafor zamówienie załatwione, będący semaforem lokalnym powoduje niezbędne opóźnienie; programista aplikacji nie ponosi odpowiedzialności za wykrycie zakończenia operacji we/wy, ale traci możliwość szybszego wykonywania procesu.

BUFOROWANIE WEJŚCIA-WYJŚCIA

- ✖ Proces powtarzający wielokrotnie operacje na tym samym strumieniu jest ciągle wstrzymywany i przełączany w stan niewykonywalny i w związku z tym traci się dużo czasu, ponieważ operacje te są bardzo kosztowne.
- ✖ Należy spowodować, aby operacje we/wy odbywały się, w miarę możliwości, bez spowalniania procesu.
- ✖ Rozwiązaniem tego problemu może być przesyłanie danych z wyprzedzeniem lub opóźnieniem, czyli buforowanie.

BUFOROWANIE WEJŚCIA-WYJŚCIA

Buforowanie można podzielić na:

- ✗ buforowanie wejścia
- ✗ buforowanie wyjścia
- ✗ buforowanie podwójne
 - ✗ dwa bufore pracują na zmianę
- ✗ buforowanie wielokrotne
 - ✗ usuwa zatory w dostępie do urządzeń we/wy

BUFOROWANIE WEJŚCIA-WYJŚCIA

Podczas używania operacji buforowanych

OPBUFWEWY(strumień, tryb, lokalizacja)

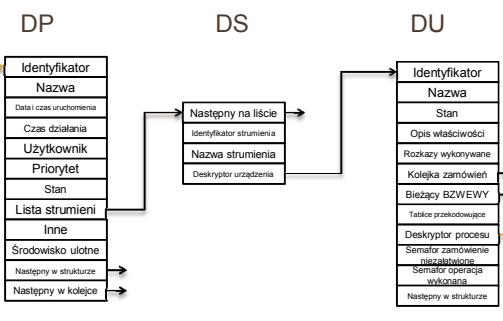
w inny sposób wykonuje się procedury we/wy.

- ✗ Operacja wejściowa obsługuje zamówienia poprzez odczytanie z bufora kolejnego dostępnego elementu.
 - ✗ Procedura generująca blok zamówień we/wy, wysyła sygnał do procesu obsługi urządzenia celem dostarczenia większej ilości danych wejściowych, wywoływaną jest jedynie w przypadku opróżnienia bufora.
- ✗ Operacja wyjściowa obsługuje zamówienia poprzez zapisanie do bufora kolejnego elementu.
 - ✗ Procedura generująca blok zamówień we/wy, wstrzymuje proces jedynie w przypadku przepelenia bufora wyjściowego.
- ✗ Operacja we/wy wraz z procesem obsługi urządzenia działają tak, jak procesy producenta i konsumenta.

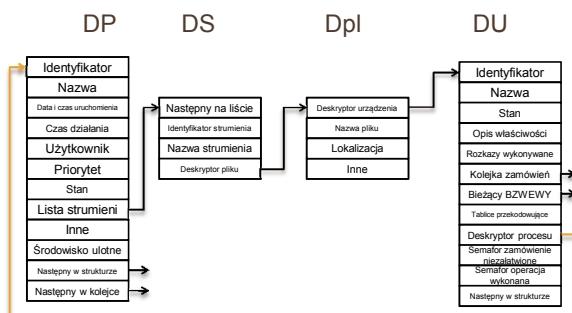
URZĄDZENIA PLIKOWE

- ✗ Założenie, iż do zapisu ze strumienia wystarczy jedynie identyfikator urządzenia docelowego, nie jest spełnione w przypadku urządzeń o dostępie swobodnym
 - ✗ dysk twardy, płyta optyczna.
- ✗ W takiej sytuacji, niezbędne jest również podanie obszaru, gdzie dany strumień się znajduje lub skąd ma zostać pobrany.
- ✗ W tego rodzaju nośnikach każdy obszar nazywany jest plikiem
 - ✗ posiada on dowolny rozmiar oraz opisującą go jednoznaczna nazwę.

POWIĄZANIE DESKRYPTORÓW – URZĄDZENIE NIEPLIKOWE



POWIĄZANIE DESKRYPTORÓW – URZĄDZENIE PLIKOWE



URZĄDZENIA PLIKOWE

- ✗ W przypadku używania urządzenia plikowego, strumień należy powiązać z nazwą pliku, nie jak poprzednio z samym urządzeniem.

define input disk2:data7

- ✗ Polecenie w takiej postaci definiuje strumień o nazwie input powiązany z plikiem nazwą data7 umieszczony na dysku disk2.

SPOOLER

- ✖ Urządzeniom niepodzielnym nie można przydzielić kilku procesów równocześnie.
- ✖ Procesy chcące korzystać z urządzeń niepodzielnych muszą czekać na zwolnienie kolejki.
- ✖ W danym momencie może być dużo zgłoszeń do urządzenia, podczas gdy później może ono stać bezczynne.
- ✖ Wskazane jest równomierne rozłożenie zamówień i przeciwdziałanie powstawaniu wąskich gardet.
- ✖ Rozwiążaniem może być zastosowanie pracy pośredniej. Dane będą przesyłane na nośnik pośredni, jakim najczęściej jest dysk.
- ✖ Następnie proces systemowy przesyła dane z plików na dysku na urządzenie; nazywa się on spoolerem (simultaneous peripheral operation on-line).

(C) IISI d.KIK PCz 2013

Systemy operacyjne

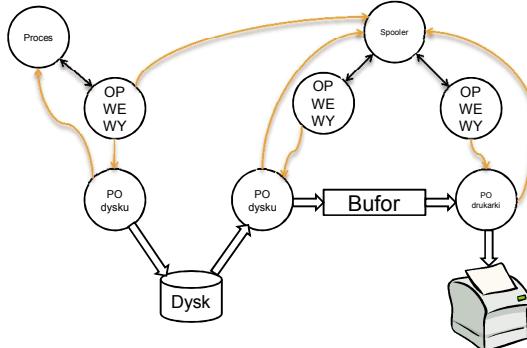
37

(C) IISI d.KIK PCz 2013

Systemy operacyjne

38

SPOOLER



(C) IISI d.KIK PCz 2013

Systemy operacyjne

39

(C) IISI d.KIK PCz 2013

Systemy operacyjne

40

SPOOLER

Działanie spoolera:

- ✖ sprawdzenie czy jest coś w kolejce
- ✖ wybranie pliku i otwarcie go
- ✖ wykonywanie transmisji do urządzenia do czasu napotkania końca pliku
- ✖ zasygnalizowanie zakończenia przetwarzania pliku

SPOOLER

- ✖ Spooler może wybierać odpowiednie pliki według zasady – FIFO lub dowolnego innego algorytmu.
- ✖ Pozwala przyspieszyć działanie procesów.
- ✖ Czyni zasób niepodzielny podzielnym.
- ✖ Zabiera zasoby dyskowe, czas procesora i pamięć.
- ✖ Nie można go stosować w systemach czasu rzeczywistego (operacje we/wy muszą w tym przypadku być wykonywane natychmiast).

(C) IISI d.KIK PCz 2013

Systemy operacyjne

41

(C) IISI d.KIK PCz 2013

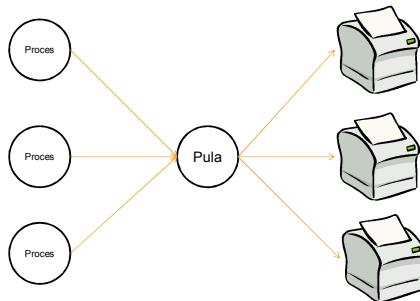
Systemy operacyjne

42

PULA

- ✖ Łączy kilka identycznych urządzeń w jedno.
- ✖ Procesy transmitują dane do puli.
- ✖ Pula przesyła dane do wybranego przez siebie urządzenia.
- ✖ Zastosowanie puli zwiększa przepustowość urządzeń.
- ✖ Proces nie musi wybierać sam urządzenia i narażać się na oczekiwanie w przypadku zajętości danego urządzenia.
- ✖ Pula może współpracować ze spoolerem.

PULA DRUKAREK

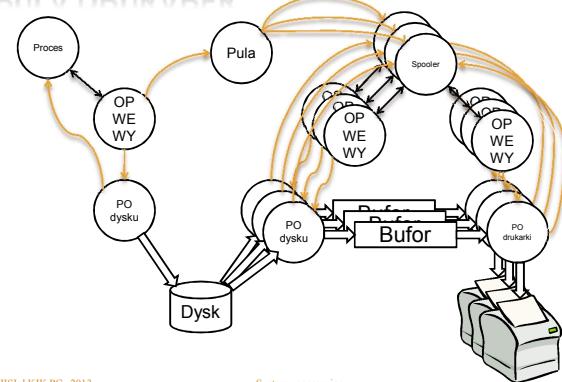


(C) IISI d.KIK PCz 2013

Systemy operacyjne

43

PULA DRUKAREK



(C) IISI d.KIK PCz 2013

Systemy operacyjne

44

MACIERZE DYSKOWE

Macierze dyskowe RAID (redundant arrays of inexpensive disks) są stosowane w celu:

- ✗ zwiększenia wydajności i
- ✗ poprawienia niezawodności przechowywania danych

(C) IISI d.KIK PCz 2013

Systemy operacyjne

45

MACIERZE DYSKOWE

Macierze dyskowe RAID występują w kilku wersjach:

- ✗ RAID0
- ✗ RAID1
- ✗ RAID2
- ✗ RAID3
- ✗ RAID4
- ✗ RAID5
- ✗ RAID6

oraz jako połączenia powyższych poziomów

(C) IISI d.KIK PCz 2013

Systemy operacyjne

46

MACIERZE DYSKOWE

Macierze dyskowe mogą być realizowane w sposób:

- ✗ sprzętowy
 - ✗ korzystając ze specjalnych kontrolerów (drogich!)
- ✗ programowy
 - ✗ przez oprogramowanie systemu operacyjnego

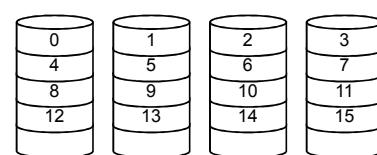
(C) IISI d.KIK PCz 2013

Systemy operacyjne

47

MACIERZE DYSKOWE – RAID0

- ✗ Przestrzeń dyskowa podzielona jest na paski.
- ✗ Paski są umieszczane cyklicznie na kolejnych dyskach.
- ✗ Zwiększenie szybkości dostępu do dysku
- ✗ Brak zabezpieczenia danych na wypadek awarii dysku



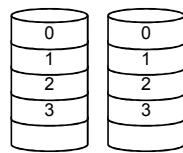
(C) IISI d.KIK PCz 2013

Systemy operacyjne

48

MACIERZE DYSKOWE – RAID1

- ✗ Przestrzeń dyskowa podzielona jest na paski.
- ✗ Paski są kopiowane na kolejnych dyskach (kopia lustrzana).
- ✗ Zwiększenie szybkości odczytu dostępu do dysku.
- ✗ Zabezpieczenie danych na wypadek awarii dowolnego dysku.
- ✗ Połowa pojemności dostępna dla systemu.



(C) IISI d.KIK PCz 2013

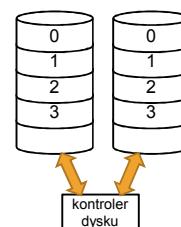
Systemy operacyjne

49

MACIERZE DYSKOWE – RAID1

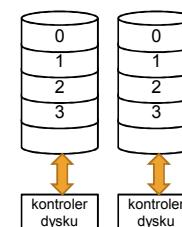
Może występować w dwóch odmianach:

Mirroring



(C) IISI d.KIK PCz 2013

Duplexing

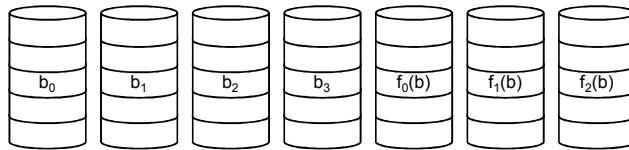


Systemy operacyjne

50

MACIERZE DYSKOWE – RAID2

- ✗ Paski są bardzo małe.
- ✗ Stosuje się kod nadmiarowy (korygujący) (f) pozwalający skorygować jeden błąd i wykryć drugi.



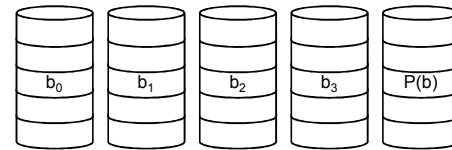
(C) IISI d.KIK PCz 2013

Systemy operacyjne

51

MACIERZE DYSKOWE – RAID3

- ✗ Parzystość z przeplotem bitów.
- ✗ Paski są bardzo małe.
- ✗ Błędy wykrywają same dyski na podstawie sum CRC.
- ✗ Stosuje się parzystość pozwalającą skorygować błąd na jednym dysku.



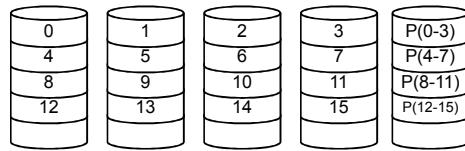
(C) IISI d.KIK PCz 2013

Systemy operacyjne

52

MACIERZE DYSKOWE – RAID4

- ✗ Parzystość z przeplotem bloków.
- ✗ Paski są wielkością bloków.
- ✗ Błędy wykrywają same dyski na podstawie sum CRC.
- ✗ Stosuje się parzystość pozwalającą skorygować błąd na jednym dysku.



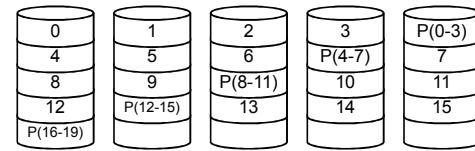
(C) IISI d.KIK PCz 2013

Systemy operacyjne

53

MACIERZE DYSKOWE – RAID5

- ✗ Parzystość rozproszona z przeplotem bloków.
- ✗ Paski są wielkością bloków.
- ✗ Błędy wykrywają same dyski na podstawie sum CRC.
- ✗ Stosuje się parzystość pozwalającą skorygować błąd na jednym dysku.



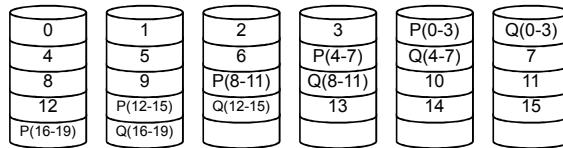
(C) IISI d.KIK PCz 2013

Systemy operacyjne

54

MACIERZE DYSKOWE – RAID6

- ✗ Parzystość rozproszona z przeplotem bloków i podwójną nadmiarowością.
- ✗ Paski są wielkością bloków.
- ✗ Błędy wykrywają same dyski na podstawie sum CRC.
- ✗ Stosuje się dwie funkcje pozwalające skorygować błąd na dwu dyskach.



(C) IISI d.KIK PCz 2013

Systemy operacyjne

55

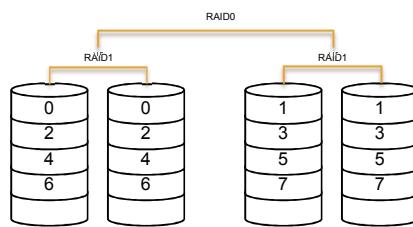
(C) IISI d.KIK PCz 2013

Systemy operacyjne

56

MACIERZE DYSKOWE – RAID 1+0

Dwa zestawy dysków lustrzanych pracują w RAID0



(C) IISI d.KIK PCz 2013

Systemy operacyjne

57

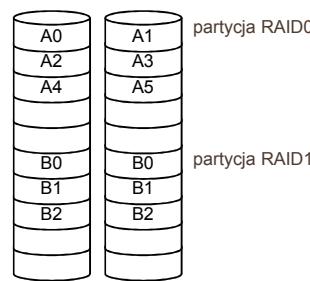
(C) IISI d.KIK PCz 2013

Systemy operacyjne

58

MACIERZE DYSKOWE – MATRIX RAID

Część przestrzeni dyskowej pracuje w RAID0, a część w RAID1.



(C) IISI d.KIK PCz 2013

Systemy operacyjne

59

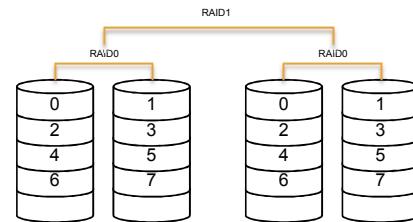
(C) IISI d.KIK PCz 2013

Systemy operacyjne

60

MACIERZE DYSKOWE – RAID 0+1

Lustrzanie pracują dwa zestawy dysków w RAID0



(C) IISI d.KIK PCz 2013

Systemy operacyjne

56

MACIERZE DYSKOWE – RAID 5+

- ✗ RAID 5+0 - Dwa zestawy RAID5 pracują w RAID0
- ✗ RAID 5+1 - Dwa zestawy RAID5 pracują w RAID1

PAMIĘĆ PODRĘCZNA DYSKU

Realizowana jest:

- ✗ w pamięci operacyjnej systemu
 - ✗ w postaci bufora o zmiennym rozmiarze
 - ✗ zarządzana przez system operacyjny
- ✗ jako dodatkowa pamięć wbudowana w napęd dysku
 - ✗ rozmiar jest ustalony przez producenta dysku
 - ✗ zarządza nią kontroler wbudowany do dysku twardego.

SYSTEM PLIKÓW

POTRZEBY Z JAKICH WYNIKA SYSTEM PLIKÓW

Pamięć dostępna bezpośrednio

- ✖ możliwość przechowywania programów lub danych wewnątrz systemu komputerowego (wygoda dla użytkownika systemu)
- ✖ w systemach wielodostępnych urządzeniem we/wy jest tylko terminal interakcyjny
- ✖ także w systemach wsadowych większość informacji (wprowadzanej przez wolniejsze terminale) przechowuje się w pamięci dostępnej bezpośrednio

(C) IISI d.KIK PCz 2013

Systemy operacyjne

1

(C) IISI d.KIK PCz 2013

Systemy operacyjne

2

POTRZEBY Z JAKICH WYNIKA SYSTEM PLIKÓW

Korzystanie ze wspólnych informacji

- ✖ w systemach ogólnego przeznaczenia:
 - + użytkownicy korzystają ze wspólnych programów i danych,
 - + instalowane są zbiory programów bibliotecznych:
 - ✗ edytory tekstu, kompilatory języków programowania, procedury użytkowe, ...
- ✖ w systemach przetwarzania transakcji:
 - + wiele procesów może mieć wspólną bazę danych

(C) IISI d.KIK PCz 2013

Systemy operacyjne

3

(C) IISI d.KIK PCz 2013

Systemy operacyjne

4

DYSKI TWARDE

IBM 350
50 24" taśmy, 4,4 MB, 1956



IBM 1311
6 14" taśmy, 2 MB, wyniesienie, 1962



IBM 3336
11 taśm, 200 MB, wyniesienie, 1973



IBM 3380
2,52 GB, 1980/81
ok. 100 000\$

IBM 3390
22 GB, 1989
ok. 80 000\$

(C) IISI d.KIK PCz 2013

PAMIĘĆ DŁUGOOKRESOWA

stosowana w postaci pamięci pomocniczej, masowej korzysta z:

- ✖ dysków magnetycznych
- ✖ taśmy magnetycznej
- ✖ płyt optycznych
- ✖ ...

(C) IISI d.KIK PCz 2013

Systemy operacyjne

4



DYSKI TWARDE

ST-506
5,25, 5 MB, 1980

ST-225
5,25, 20 MB



3,5"
2,5" SSD
2,5, 250 GB



(C) IISI d.KIK PCz 2013

Systemy operacyjne

6

DYSKI TWARDE

8", 5.25",
3.5", 2.5",
1.8", 1"

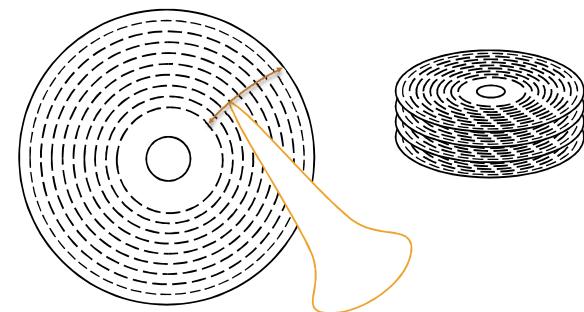
5.25"
pełna
wysokość i
2.5"

(C) IISI d.KIK PCz 2013



7

DYSKI TWARDE - BUDOWA



(C) IISI d.KIK PCz 2013

Systemy operacyjne

8

ADRESACJA SEKTORÓW

- ✖ Adresacja trójwymiarowa: CHS
- ✖ Adresacja liniowa: LBA

Numer sektora w adresacji LBA można obliczyć przy pomocy wzoru:

$$LBA = (CYL * LGŁC + GL)* LSS + SEK - 1.$$

gdzie:
 CYL – nr cylinda,
 LGŁC – liczba głowic na cylinderze,
 GL – liczba sektorów na ścieżce,
 LSS – liczba sektorów na ścieżce,
 SEK – nr sektora

Aby obliczyć z adresu LBA adres CHS należy posłużyć się wzorami :

$$CYL = LBA / (LGŁC * LSS)$$

$$X = LBA \% (LGŁC * LSS)$$

$$GL = X / LSS$$

$$SEK = X \% LSS + 1$$

(C) IISI d.KIK PCz 2013

Systemy operacyjne

9

SYSTEM PLIKÓW

- ✖ tworzy organizację danych w pamięci masowej i pozwala z nich korzystać w sposób wygodny dla użytkownika,
- ✖ udostępnia dane w postaci plików,
- ✖ ukrywa szczegóły dostępu do urządzenia,
- ✖ ukrywa budowę organizacji danych.

(C) IISI d.KIK PCz 2013

Systemy operacyjne

10

PLIK

Plik jest zbiorem danych traktowanych jak pewna całość;

- ✖ jest jednostką logiczną dostępu do danych,
- ✖ system wykonuje na nim pewne operacje,
- ✖ zapisywany jest w blokach nośnika (512-4096).

(C) IISI d.KIK PCz 2013

Systemy operacyjne

11

PLIK

- ✖ Format pliku może być:
- ✖ dokładnie określony
 - ✖ pliki baz danych
- ✖ swobodny
 - ✖ pliki tekstowe

(C) IISI d.KIK PCz 2013

Systemy operacyjne

12

PLIK - OPERACJE

- ✖ tworzenie
- ✖ usuwanie
- ✖ zapisywanie
- ✖ czytanie
- ✖ dopisywanie
- ✖ skracanie
- ✖ zmiana pozycji w pliku
- ✖ przemianowywanie
- ✖ otwieranie
- ✖ zamknięcie

(C) IISI d.KIK PCz 2013

Systemy operacyjne

13

PLIK - METODY DOSTĘPU

- ✖ dostęp sekwencyjny
- ✖ dostęp bezpośredni (swobodny)
- ✖ dostęp indeksowy
 - ✖ wymaga dodatkowego pliku indeksu

(C) IISI d.KIK PCz 2013

Systemy operacyjne

14

ZADANIA SYSTEMU PLIKÓW:

- ✖ tworzenie i usuwanie plików
- ✖ dostęp do plików w celu czytania i pisania
- ✖ zarządzanie przestrzenią pamięci pomocniczej
 - ✖ użytkownika nie powinno obchodzić dokładne miejsce przechowywania pliku w pamięci pomocniczej
- ✖ odwoływanie się do plików za pomocą nazw symbolicznych

(C) IISI d.KIK PCz 2013

Systemy operacyjne

15

ZADANIA SYSTEMU PLIKÓW CD:

- ✖ ochrona plików przed skutkami uszkodzenia systemu
 - ✖ użytkownicy nieczęściej powierzą cokolwiek systemowi, jeśli nie będą przekonani o jego niezawodności
- ✖ powinien pozwalać na współużywalność plików
- ✖ ochrona plików przed nieuprawnionym dostępem

(C) IISI d.KIK PCz 2013

Systemy operacyjne

16

KATALOGI PLIKÓW

to tabele zawierające wpisy z informacjami o plikach:

- ✖ nazwa pliku
- ✖ adres pliku w pamięci pomocniczej
- ✖ typ pliku (tekstowy, binarny)
- ✖ atrybuty - kontrola dostępu
- ✖ inf. administracyjne - np. data i czas

(C) IISI d.KIK PCz 2013

Systemy operacyjne

17

KATALOGI PLIKÓW

- ✖ jednopoziomowy
- ✖ główny, użytkownika
- ✖ wielopoziomowy

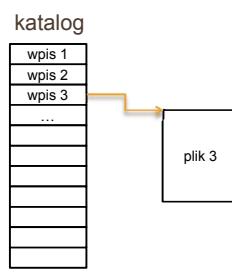
(C) IISI d.KIK PCz 2013

Systemy operacyjne

18

KATALOGI PLIKÓW

jednopoziomowy



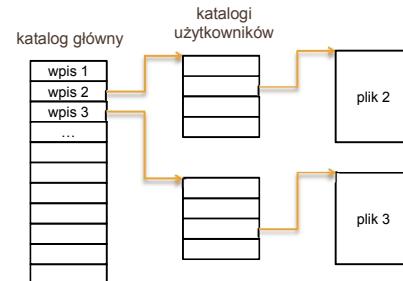
(C) IISI d.KIK PCz 2013

Systemy operacyjne

19

KATALOGI PLIKÓW

główny, użytkownika



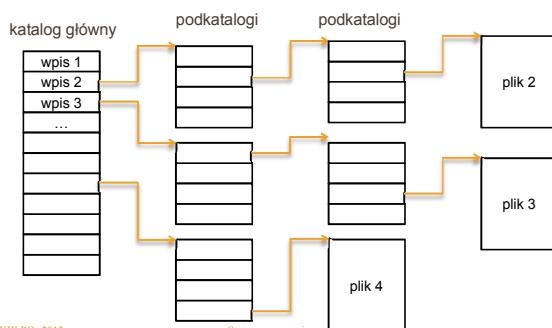
(C) IISI d.KIK PCz 2013

Systemy operacyjne

20

KATALOGI PLIKÓW

wielopoziomowy



(C) IISI d.KIK PCz 2013

Systemy operacyjne

21

KATALOGI PLIKÓW

- ✖ w katalogu nazwy plików muszą być unikalne
- ✖ nazwy mogą powtarzać się w różnych podkatalogach
- ✖ pełna nazwa pliku składa się z nazw kolejnych podkatalogów począwszy od korzenia (katalogu głównego) oraz z nazwy pliku
 - ✖ /home/kowalski/dokumenty/list.txt
- ✖ jeden z katalogów można uczynić bieżącym, wówczas zamiast podawania pełnej nazwy wystarczy sama nazwa pliku
 - ✖ list.txt
- ✖ wydajność przeglądania można zwiększyć stosując sortowanie wpisów lub b-drzewa

(C) IISI d.KIK PCz 2013

Systemy operacyjne

22

WSPÓŁUŻYTKOWANIE I OCHRONA INFORMACJI

Pliki w systemie wielodostępnym mogą być współużytkowane. Trzeba wprowadzić metody ochrony danych przed niepowołanym dostępem:

- ✖ maska ochrony pliku
- ✖ lista kontroli dostępu
- ✖ dowiązania

(C) IISI d.KIK PCz 2013

Systemy operacyjne

23

MASKA OCHRONY PLIKU

Maska ochrony pliku może określać prawa dostępu dla różnych klas użytkowników:

- ✖ O- właściciel (owner)
- ✖ G- członkowie grupy lub partnerzy (group)
- ✖ W- inni (world)

Typowe prawa dostępu:

- ✖ R- prawo czytania (read)
- ✖ W- prawo pisania (write)
- ✖ E/X- prawo wykonywania (execute)
- ✖ D- prawo kasowania (delete)

(C) IISI d.KIK PCz 2013

Systemy operacyjne

24

MASKA OCHRONY PLIKU

- ✖ Właściciel może określić maskę ochrony:
 - ✖ np.: O:RWED ,G:RW ,W:R
 - ✖ która oznacza, że właściciel pliku ma wszystkie prawa dostępu, jego partnerzy mogą plik czytać i zapisywać, a wszyscy pozostali mogą jedynie czytać.
- ✖ Właściciele czasami ograniczają sobie prawa dostępu, aby zabezpieczyć się przed usunięciem pliku (utratą ważnych danych).
- ✖ Właściciel może zmieniać maski ochrony plików, które do niego należą.

(C) IISI d.KIK PCz 2013

Systemy operacyjne

25

MASKA OCHRONY PLIKU

- ✖ System rozpoznaje użytkownika po kodzie identyfikacji użytkownika, sprawdza, czy jest on właścicielem oraz czy należy do określonej grupy następnie nadaje mu prawa do pliku.
- ✖ Użytkownicy najczęściej mogą należeć do wielu grup.
- ✖ W systemie Unix/Linux maska ochrony pliku ma postać:
 - ✖ rwx rwx rwx

(C) IISI d.KIK PCz 2013

Systemy operacyjne

26

LISTA KONTROLI DOSTĘPU

- ✖ jest dołączona do pliku lub katalogu i zawiera pełną informację o tym, którzy użytkownicy mają dostęp (i jaki) do określonego pliku.
- ✖ Każda pozycja na liście kontroli dostępu zawiera następujące informacje:
 - ✖ określenie użytkownika lub grupy użytkowników, której dotyczy dana pozycja listy kontroli dostępu
 - ✖ definicja praw dostępu udzielanych w danej pozycji listy kontroli dostępu np.:RWED
 - ✖ opcje powiązane z dana pozycja listy kontroli dostępu: są to dodatkowe możliwości np. dziedziczenie praw.

(C) IISI d.KIK PCz 2013

Systemy operacyjne

27

LISTA KONTROLI DOSTĘPU

- ✖ jest bardziej elastyczna niż maska ochrony pliku,
- ✖ występuje w systemach VMS oraz Windows NT+,
- ✖ powoduje wolniejsze otwieranie pliku, gdyż trzeba sprawdzić wszystkie pozycje na liście.

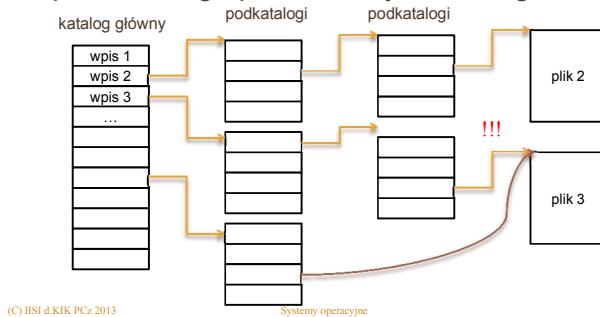
(C) IISI d.KIK PCz 2013

Systemy operacyjne

28

DOWIAZANIA

polegają na umożliwieniu odwoływania się do współdzielonego pliku z różnych katalogów



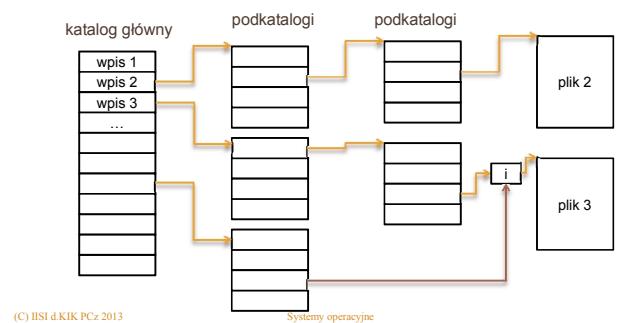
(C) IISI d.KIK PCz 2013

Systemy operacyjne

29

DOWIAZANIA TRWAŁE

zostaje dodana struktura i-węzła

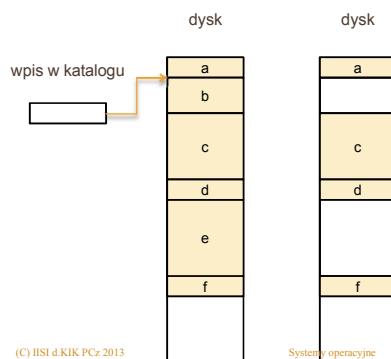


(C) IISI d.KIK PCz 2013

Systemy operacyjne

30

SYSTEM PLIKÓW ZWARTYCH (PRZYDZIAŁ CIĄGŁY)



(C) IISI d.KIK PCz 2013

37

SYSTEM PLIKÓW ZWARTYCH (PRZYDZIAŁ CIĄGŁY)

Zalety:

- ✖ szybkość dostępu
- ✖ brak dodatkowych struktur danych
- ✖ łatwość odzyskania danych po awarii

Wady:

- ✖ fragmentacja pamięci
- ✖ konieczność upakowywania pamięci

(C) IISI d.KIK PCz 2013

Systemy operacyjne

38

ŁAŃCUCH POWIAZANYCH BLOKÓW (PRZYDZIAŁ LISTOWY)

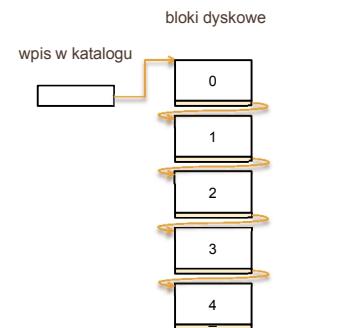
- ✖ Kilka bajtów każdego bloku w pliku służą jako wskaźnik do następnego bloku. Ostatni blok zawiera wskaźnik pusty (na ogół 0).
- ✖ Pozycja w katalogu zawiera informacje o pierwszym (ostatnim) bloku pliku.
- ✖ Dostęp do pliku jest z sekwencyjny, ponieważ do każdego bloku można jedynie dotrzeć posuwając się w dół wzdłuż łańcucha.

(C) IISI d.KIK PCz 2013

Systemy operacyjne

39

ŁAŃCUCH POWIAZANYCH BLOKÓW (PRZYDZIAŁ LISTOWY)



(C) IISI d.KIK PCz 2013

Systemy operacyjne

40

ŁAŃCUCH POWIAZANYCH BLOKÓW (PRZYDZIAŁ LISTOWY)

- ✖ Ta metoda tworzenia łańcucha powiązanych ze sobą bloków jest najodpowiedniejsza w przypadku sekwencyjnego przetwarzania plików
 - ✖ dodatkowy koszt dostępu do pliku sprowadza się do przeczytania kolejnych bloków.
- ✖ Metoda ta jest mało elastyczna, skutki uszkodzenia jednego bloku (i znajdujących się w nim powiązań) mogą niespodziewanie rozszerzyć się na cały system plików.
- ✖ Można zmniejszyć to niebezpieczeństwo zwiększąc zajęte dodatkowo obszary pamięci i zapamiętać nadmiarowe dowiązania odwrotne.

(C) IISI d.KIK PCz 2013

Systemy operacyjne

41

MAPA PLIKÓW (TABLICA PRZYDZIAŁU PLIKÓW)

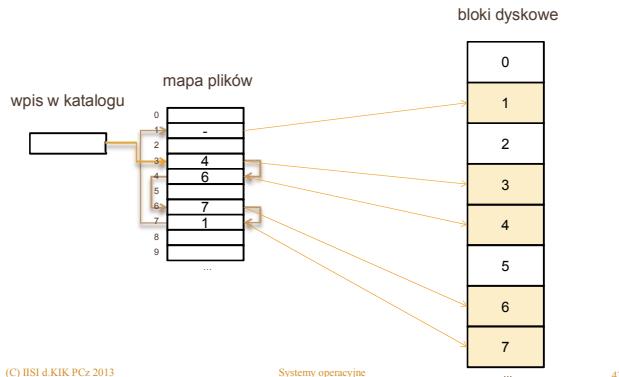
- ✖ Stan dysku jest zapisywany w mapie plików, nazywanej tablicą przydziału plików, w której każdy blok na dysku jest reprezentowany przez jeden wpis w mapie.
- ✖ Pozycja pliku w katalogu użytkownika wskazuje na numer pierwszego bloku pliku; element mapy o tym samym numerze wskazuje na kolejny element mapy/blok pliku itd. Ostatni blok w pliku jest reprezentowany przez wskaźnik pusty.

(C) IISI d.KIK PCz 2013

Systemy operacyjne

42

MAPA PLIKÓW (TABLICA PRZYDZIAŁU PLIKÓW)



(C) IISI d.KIK PCz 2013

Systemy operacyjne

43

MAPA PLIKÓW (TABLICA PRZYDZIAŁU PLIKÓW)

- ✖ Zajętość obszaru pamięci pomocniczej zależy od rozmiarów elementów mapy (12-32bitów) oraz wielkości pamięci pomocniczej.
- ✖ W elementach mapy plików mogą się znajdować dodatkowe informacje nadmiarowe takie jak:
 - ✖ numer identyfikacyjny pliku, który może się przydać do odtwarzania pliku po uszkodzeniu systemu.
- ✖ Dostęp do elementów mapy plików jest sekwencyjny, co zwalnia dostęp do pliku.
- ✖ W celu ułatwienia rozszerzania i kasowania pliku, w pozycjach katalogu użytkownika można umieścić wskaźnik do ostatniego elementu mapy.

(C) IISI d.KIK PCz 2013

Systemy operacyjne

44

MAPA PLIKÓW (TABLICA PRZYDZIAŁU PLIKÓW)

- ✖ Przeczytanie pliku złożonego z N bloków wymaga N dodatkowych dostępów do dysku w celu przeczytania odpowiednich części mapy plików,
 - ✖ można to ograniczyć rozmieszczać pliki w sąsiednich blokach tak, by jeden odczyt z mapy wystarczał na wiele bloków.
- ✖ Aby zapobiec utracie danych należy zapamiętać co najmniej 2 kopie mapy i to najlepiej w różnych rejonach dysku, aby awaria sprzętu nie spowodowała zniszczenia wszystkich kopii.

(C) IISI d.KIK PCz 2013

Systemy operacyjne

45

BLOKI INDEKSÓW (PRZYDZIAŁ INDEKSOWY)

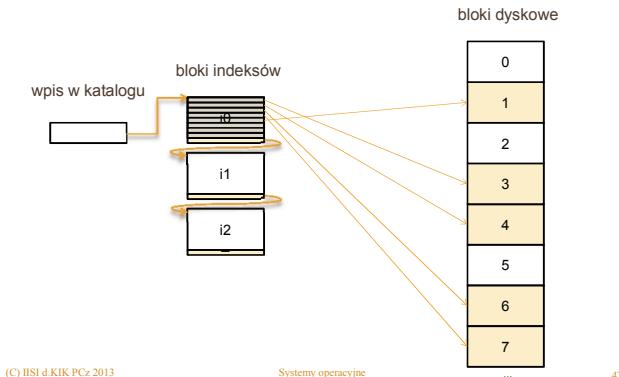
- ✖ Wskaźniki dowiązań do każdego pliku są pamiętane w odrębnych blokach indeksów na dysku.
- ✖ Dla dużego pliku trzeba przeznaczyć kilka bloków indeksów powiązanych w łańcuch, każdy blok indeksów musi mieć wskaźnik do następnego bloku.
- ✖ Pozycja w katalogu wskazuje na pierwszy blok w łańcuchu bloków indeksów.

(C) IISI d.KIK PCz 2013

Systemy operacyjne

46

BLOKI INDEKSÓW (PRZYDZIAŁ INDEKSOWY)



(C) IISI d.KIK PCz 2013

Systemy operacyjne

47

BLOKI INDEKSÓW (PRZYDZIAŁ INDEKSOWY)

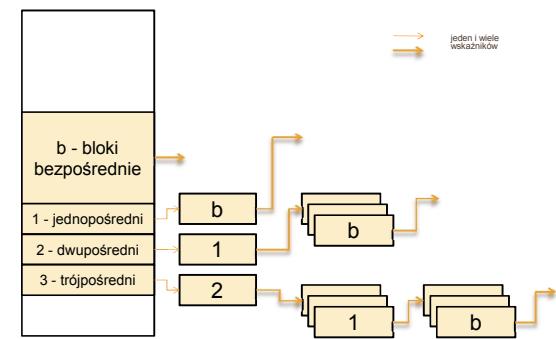
- ✖ Największą zaletą bloków indeksów jest to, że dostęp do pliku nie musi być sekwencyjny, ale swobodny (określenie nazwy pliku i odległości w bloku indeksów)
- ✖ Uszkodzenie bloków indeksów powoduje poważną utratę danych, aby do tego nie dopuścić można przechowywać się kilka kopii bloków indeksów w różnych obszarach dysku;
 - ✖ duża strata miejsca lecz opłacalne ze względu na bezpieczeństwo danych.

(C) IISI d.KIK PCz 2013

Systemy operacyjne

48

PRZYDZIAŁ INDEKSOWY – I-WĘZEŁ



(C) IISI d.KIK PCz 2013

49

ZARZĄDZANIE WOLNĄ PRZESTRZENIĄ

System musi wiedzieć, które bloki są wolne. W tym celu stosuje jedną z metod:

- ✖ wektor bitowy/mapa bitowa
 - ✖ każdy bit mapy informuje o zajętości danego bloku
- ✖ lista powiązana – jak w przydziale listowym
 - ✖ system wskazuje tylko na pierwszy wolny blok, a ten na następny
- ✖ grupowanie
 - ✖ pierwszy wolny blok zawiera adresy innych wolnych, z których ostatni zawiera wskazania na kolejne
- ✖ zliczanie
 - ✖ system pamięta numer pierwszego wolnego bloku i liczbę wolnych następujących po nim

(C) IISI d.KIK PCz 2013

Systemy operacyjne

50

SYSTEMY PLIKÓW ZE STRUKTURĄ DZIENNIKA

- ✖ nazywane również: transakcyjnymi systemami plików ze strukturą dziennika lub systemami plików z księgowaniem (kronikowaniem);
- ✖ wszystkie zmiany zapisuje się najpierw w dzienniku, po ich zatwierdzeniu system aktualizuje struktury systemu plików i usuwa wykonaną transakcję z dziennika;
- ✖ w przypadku awarii transakcje można dokonczyć na podstawie dziennika, tracone są wyłącznie dane nie zatwierdzone;
- ✖ systemy z księgowaniem:
 - ✖ NTFS, ext3, JFS, XFS, ReiserFS, Reiser4, BeFS, ...

(C) IISI d.KIK PCz 2013

Systemy operacyjne

51

OGÓLNA BUDOWA PARTYCJI FAT



- ✖ Opis systemu znajduje się w BS
- ✖ Elementy mapy plików mogą być 12, 16 lub 32 bitowe
- ✖ Podkatalogi mają postać plików (niewidocznych jako pliki) i mogą znajdować się w dowolnym miejscu i mieć dowolny rozmiar.

(C) IISI d.KIK PCz 2013

Systemy operacyjne

52

OGÓLNA BUDOWA PARTYCJI NTFS



- ✖ MFT – Master File Table zajmuje 12.5, 25, 37.5 lub 50% partycji
- ✖ W MFT znajdują się 1 KB wpisy opisujące pliki i katalogi

(C) IISI d.KIK PCz 2013

Systemy operacyjne

53

ZAWARTOŚĆ MFT

- ✖ Pierwsze 24 wpisy są zarezerwowane dla systemu na np.: kopię MFT, mapę bitową wolnych bloków, katalog główny, definicję atrybutów i zabezpieczeń, ...

0	\$Mft
1	\$MftMirr
2	\$LogFile
3	\$Volume
4	\$AttrDef
5	\
6	\$BitMap
7	\$Boot
8	\$BadClus
9	\$Secure
10	\$UpCase
11	\$Extend
	...
24	Pliki i katalogi użytkownika

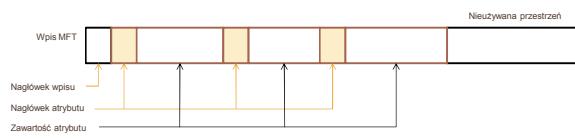
(C) IISI d.KIK PCz 2013

Systemy operacyjne

54

STRUKTURA WPISU DO MFT

- ✗ Ilość atrybutów może być różna.
- ✗ Rozmiar atrybutów może być różny.



(C) IISI d.KIK PCz 2013

Systemy operacyjne

55

SPOSÓB ADRESOWANIA KLASTRÓW W NTFS

- ✗ LCN (Logical Cluster Numbers) – numer logiczny klastra na dysku
- ✗ VCN (Virtual Cluster Numbers) – numer wirtualny klastra w pliku.
- ✗ Mapowanie klastrów VCN na LCN (1720, 1721, 1722, 1723, 1724, 1975, 1976 i 1977) odbywa się zgodnie z tabelą zawartą we wpisie MFT.

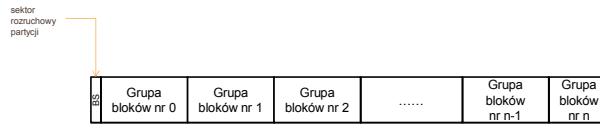
Początkowy VCN	Początkowy LCN	Długość
0	1720	5
5	1975	3

(C) IISI d.KIK PCz 2013

Systemy operacyjne

56

OGÓLNA BUDOWA PARTYCJI EXT2/EXT3



- ✗ Cała partycja podzielona jest na grupy bloków o jednakowym rozmiarze (z wyjątkiem ostatniej)

(C) IISI d.KIK PCz 2013

Systemy operacyjne

57

BUDOWA GRUPY BLOKÓW PARTYCJI EXT2/EXT3



- ✗ Wszystkie grupy zawierają identyczny superblok i tablice deskryptorów grup.
- ✗ Każda grupa posiada własne bitmapy bloków dyskowych (danej grupy) i i-węzłów (tej grupy) oraz tablicę i-węzłów

(C) IISI d.KIK PCz 2013

Systemy operacyjne

58

STRUKTURA I-WĘZŁA

NR BAJTÓW	POLE
28 - 31	dalej określone w tabelach
32 - 35	atrybuty - zapisywane w postaci flag o następującym znaczeniu: 00000000 - brak 00000001 - bezpieczne usunięcie 00000002 - kopią zapisa poziomu usunięcia 00000003 - bezpieczne zapisu 00000004 - zapis do określonego adresu 00000005 - tryb dopuszcania 00000006 - brak aktywnego zapisu, dopuszczalny dalszy zapis 00000007 - brak aktywnego zapisu, dopuszczalny dalszy zapis 00000008 - kopiowanie
36 - 39	rozsypanie
40 - 47	12 bajtów adresów bloków danych
48 - 51	adres poprzedzające pośredniego bloku danych
52 - 55	adres poprzedniego pośredniego bloku danych
56 - 59	adres poprzedniego pośredniego bloku danych
60 - 103	numer plików
104 - 107	adresy bloków rozsypanych arkuszy
108 - 111	numer plików (stanowiące 32 bajty) - adresy kolejnych arkuszy NCL dla katalogu
112 - 115	adresy bloku fragmentu
116 - 119	indeks fragmentu
120 - 123	rozsypanie
124 - 125	jeśli GID (stanowiące 16 bajtów)
126 - 127	jeśli GID (stanowiące 16 bajtów)
128 - 129	rozsypanie

(C) IISI d.KIK PCz 2013

Systemy operacyjne

59

KATALOGI PARTYCJI EXT2/EXT3



- ✗ Wpisy zajmują różne ilości miejsca.
- ✗ Wskazują na następny wpis.
- ✗ Ostatni w katalogu wskazuje na koniec bloku.

(C) IISI d.KIK PCz 2013

Systemy operacyjne

60

NR BAJTÓW	POLE
0 - 1	numer i-węzła - odnoszący do i-węzła, na który wskazuje wpis katalogowy
4 - 5	dużo wpisów katalogowego - podana w bajtach
6	dużo nazwy pliku - podana w bajtach
7	dużo nazwy pliku (stanowiącej bajty) - typ pliku - definicje wartości pola typ pliku: 00b - nienazwany 01b - plik danych 02b - katalog 03b - urządzenie znakowe 04b - urządzenie blokowe 05b - FIFO 06b - gniazdo Unikta 07b - dowsiadanie symboliczne
8 - X	nazwa pliku

SYSTEM NFS (NETWORK FILE SYSTEM)

- ✗ Sieciowy system plików umożliwia korzystanie z zasobów plikowych innych komputerów jak z zasobów lokalnych.
- ✗ Realizowany jako system klient serwer.
- ✗ Zdalny zasób musi być udostępniony.
- ✗ Montuje się go w lokalnym katalogu.
 - ✗ Niektóre systemy zezwalają na montowanie kaskadowe.
 - ✗ Protokół montowania służy do nawiązania połączenia między klientem a serwerem. Zwraca uchwyt plikowy.
- ✗ Protokół NFS dostarcza zbiór wywołań procedur zdalnych do operacji plikowych:
 - ✗ szukanie pliku, czytanie katalogu, manipulowanie dowiązaniami i katalogami, dostęp do atrybutów pliku, czytanie i pisanie, otwieranie i zamknięcie pliku.
- ✗ System przeszukuje ścieżkę dostępu do pliku i odnajduje go we właściwych zasobach (lokalnych lub zdalnych).

(C) IISI d.KIK PCz 2013

Systemy operacyjne

61

(C) IISI d.KIK PCz 2013

Systemy operacyjne

62

DESKRYPTOR PLIKU

centralny

nazwa pliku
adres deskryptora urządzenia przechowującego plik
lokalizacja pierwszego bloku na urządzeniu
licznik użycia pliku
bit pisania pliku

lokalny

lokalizacja następnego bloku
tryb dostępu

Deskryptor lokalny jest przypisany do procesu.

Deskryptor centralny jest przypisany do urządzenia.

(C) IISI d.KIK PCz 2013

Systemy operacyjne

63

(C) IISI d.KIK PCz 2013

Systemy operacyjne

64

BEZPIECZEŃSTWO SYSTEMU PLIKÓW

Aby zapewnić bezpieczeństwo systemu plików stosowane są między innymi:

- ✗ macierze dyskowe RAID (Redundant Array Inexpensive Disks - nadmiarowa macierz niedrogich dysków)
- ✗ składowanie (ang. back up)
- ✗ serwery lustrzane

(C) IISI d.KIK PCz 2013

Systemy operacyjne

65

(C) IISI d.KIK PCz 2013

Systemy operacyjne

66

SKŁADOWANIE PLIKÓW

- ✗ Polega na zapisaniu na nośniku zewnętrznym (najczęściej taśma magnetyczna lub płyta) zawartości systemu plików.
- ✗ Jest to proces czasochłonny, może trwać wiele godzin.
- ✗ Dzieli się na:
 - ✗ okresowe (globalne, pełne)
 - ✗ przyrostowe

PROCEDURA OTWIERANIA PLIKU

- ✗ wyszukiwanie nazwy pliku
- ✗ sprawdzenie, czy proces chcący otworzyć plik ma do niego prawo w określonym trybie
- ✗ sprawdzenie, czy plik jest już otwarty
- ✗ ustalenie urządzenia, na którym znajduje się plik oraz jego lokalizacji
- ✗ utworzenie deskryptora pliku
 - ✗ deskryptator centralny
 - ✗ deskryptator lokalny

(C) IISI d.KIK PCz 2013

Systemy operacyjne

61

(C) IISI d.KIK PCz 2013

Systemy operacyjne

62

SKŁADOWANIE PLIKÓW

- ✖ **Okresowe** - polega na zapisaniu na nośniku całego systemu plików.
 - ✖ Proces jest czasochłonny.
- ✖ Odzyskanie uszkodzonych danych polega na przywróceniu danych z ostatniej kopii.

(C) IISI d.KIK PCz 2013

Systemy operacyjne

67

(C) IISI d.KIK PCz 2013

Systemy operacyjne

68

SKŁADOWANIE PLIKÓW

- ✖ **Przyrostowe** – na nośniku zapisuje się tylko nowe pliki oraz te, które uległy modyfikacji od ostatniego składowania.
 - ✖ Proces przebiega stosunkowo szybko.
- ✖ W przypadku awarii dane należy najpierw odtworzyć z ostatniego składowania okresowego, a następnie z kolejnych kopii przyrostowych.
 - ✖ Wymaga więcej czasu niż odtwarzanie z kopii okresowej.

SKŁADOWANIE PLIKÓW

Metodologia dla istotnych danych:

- ✖ 7 kopii dziennych
- ✖ 5 kopii tygodniowych
- ✖ 12 kopii miesięcznych
- ✖ kopie roczne

(C) IISI d.KIK PCz 2013

Systemy operacyjne

69

(C) IISI d.KIK PCz 2013

Systemy operacyjne

70

SERWERY LUSTRZANE

- ✖ Zestaw identycznych serwerów połączonych szybkimi łączami.
- ✖ Jeden podstawowy (działa na bieżąco)
- ✖ Co najmniej jeden zapasowy (przechowuje kopię podstawowego)
- ✖ W przypadku awarii serwera podstawowego następuje przełączenie na jeden z zapasowych.
- ✖ Serwery odległe o co najmniej kilkadziesiąt kilometrów, umieszczone w miejscach strategicznie nieistotnych.

PLANOWANIE DOSTĘPU DO DYSKU

- ✖ W systemie wieloprogramowym w kolejce do dysku może oczekивать więcej niż jedno żądanie.
- ✖ Istnieje kilka strategii szeregowania żądań dyskowych.
- ✖ FCFS
- ✖ SSTF
- ✖ SCAN
- ✖ C-SCAN
- ✖ LOOK
- ✖ C-LOOK

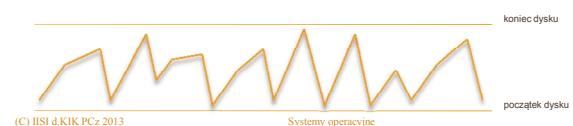
(C) IISI d.KIK PCz 2013

Systemy operacyjne

71

FCFS (FIRST-COME, FIRST-SERVED)

- ✖ Pierwszy przyszedł pierwszy zostanie obsłużony (obsługa w kolejności pojawiania się żądań).
 - ✖ prosta implementacja
 - ✖ akceptowalna przy małym obciążeniu, przy dużym długi czas dostępu



SSTF (SHORTEST SEEK TIME FIRST)

- ✖ Do obsługi wybiera żądanie z najmniejszym czasem przeszukiwania względem bieżącej pozycji głowicy.
- ✖ przepustowość lepsza niż przy FCFS, średni czas obsługi krótszy dla średniego obciążenia
- ✖ nie jest optymalny
- ✖ możliwe zagłodzenie, skrajne ścieżki są dyskryminowane
- ✖ akceptowalny w systemach wsadowych, ale nie akceptowalny w systemach interakcyjnych

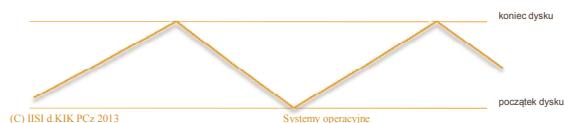


(C) IISI d.KIK PCz 2013

73

SCAN (ALGORYTM WINDY)

- ✖ Głowica przemieszcza się od brzegu do brzegu dysku obsługując napotkane po drodze żądania.
- ✖ z chwilą gdy głowica zmienia kierunek ruchu będzie stosunkowo mało żądań do obsługi
- ✖ najczęściej spotykana w praktyce strategia
- ✖ dobra z punktu widzenia przepustowości i średniego czasu obsługi, choć skrajne ścieżki nadal nieco dyskryminowane

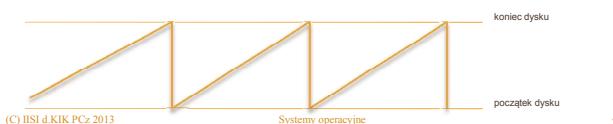


(C) IISI d.KIK PCz 2013

74

C-SCAN

- ✖ Głowica porusza się w jedną stronę, realizując zlecenia, przenosi się na początek i dopiero wtedy ponownie rozpoczyna realizację zleceń.
- ✖ większe jest prawdopodobieństwo powstania zleć na początku dysku niż w miejscu, gdzie głowica właśnie była.
- ✖ żadne ścieżki nie są dyskryminowane
- ✖ badania symulacyjne wykazały, że najlepiej połączyć SCAN (przy małym obciążeniu) z C-SCAN (przy dużym obciążeniu)



(C) IISI d.KIK PCz 2013

75

LOOK

- ✖ Wariant (praktyczny) algorytmu SCAN – głowica przesuwa się do ostatniego żądania w danym kierunku poczyn zmienia kierunek ruchu.



(C) IISI d.KIK PCz 2013

76

C-LOOK

- ✖ Wariant (praktyczny) algorytmu C-SCAN – głowica przesuwa się do ostatniego żądania w kierunku realizacji dostępu, poczyn wraca do bieżąco pierwszego żądania.



(C) IISI d.KIK PCz 2013

77

KLASTRY

- ✖ Klaster (ang. cluster - grono) jest to jednostka alokacji plików wprowadzona, aby umożliwić obsługę dysków o większych pojemnościach.
- ✖ Problem rozmiaru klastrów
 - ✖ W miarę wzrostu rozmiaru klastra wzrasta ilość marnowanego miejsca na dysku. Gdy jest zapisywany plik, nie zajmuje całego klastra, w pozostałym miejscu nie można nic zapisać. Powstaje zjawisko fragmentacji wewnętrznej.

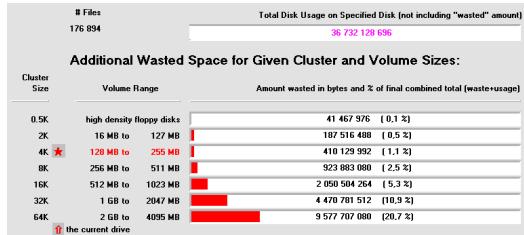
(C) IISI d.KIK PCz 2013

Systemy operacyjne

78

KLASTRY

Utrata pamięci dyskowej w zależności od rozmiaru klastra.



(C) IISI d.KIK PCz 2013

Systemy operacyjne

79

(C) IISI d.KIK PCz 2013

Systemy operacyjne

80

PODŁĄCZENIA DYSKÓW

DAS (direct attached storage) - dysk może być podłączony bezpośrednio do komputera macierzystego przy użyciu interfejsu:

- ✖ IDE (integrated drive electronics)/EIDE (enhanced IDE)/ATA(Advanced Technology Attachments)
- ✖ SATA (Serial ATA)
- ✖ SCSI (Small Computer Systems Interface)
- ✖ SAS (Serial Attached SCSI)
- ✖ FC (Fibre Channel)
- ✖ USB (Universal Serial Bus)
- ✖ ...

(C) IISI d.KIK PCz 2013

Systemy operacyjne

81

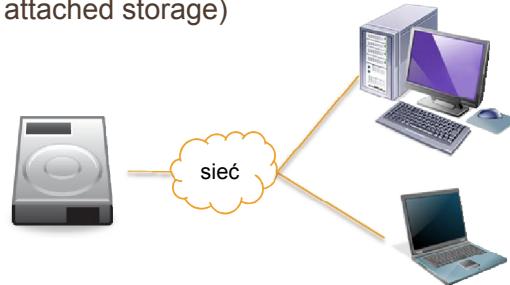
(C) IISI d.KIK PCz 2013

Systemy operacyjne

82

PODŁĄCZENIA DYSKÓW

- ✖ Dysk podłączony do sieci – NAS (network attached storage)



PODŁĄCZENIA DYSKÓW

- ✖ Sieć pamięciowa – SAN (storage-area network) jest siecią prywatną stosującą protokoły pamięci, a nie sieciowe.



(C) IISI d.KIK PCz 2013

Systemy operacyjne

83

PRZYDZIAŁ ZASOBÓW I PLANOWANIE

TECHNIKI DZIELENIA OGRANICZONEGO ZBIORU ZASOBÓW

- ✖ wzajemne wyłączanie procesów od zasobów niepodzielnych
- ✖ zapobieganie zakleszczeniom
- ✖ zapewnienie dużego wykorzystania zasobów
- ✖ dostarczenie zasobów w „rozsądnym” czasie

PRZYDZIAŁ ZASOBÓW

- ✖ mechanizmy przydziału zasobów – różnego rodzaju techniki i struktury przydziału zasobów
- ✖ polityka przydziału zasobów – zarządza sposobami korzystania z mechanizmów
 - ✖ niewłaściwe przyjmowanie zamówień może prowadzić do zakleszczenia lub przeciążenia systemu

ZAKLESZCZENIE (BLOKADA, IMPAS)

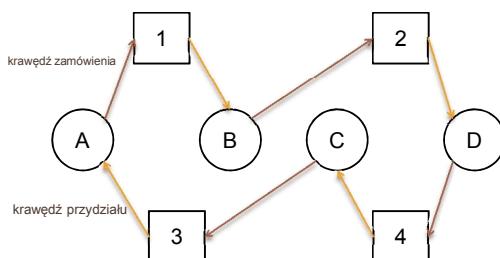
- ✖ W środowisku wieloprogramowym wiele procesów może rywalizować o skończoną liczbę zasobów.
- ✖ Zakleszczeniem nazywa się sytuację, w której pewien zbiór oczekujących procesów nie może zmienić swego stanu, ponieważ zamawiane przez nie zasoby są przetrzymywane przez inne procesy z tego zbioru.

WARUNKI KONIECZNE I WYSTARCZAJĄCE POWSTANIA ZAKLESZCZENIA

Do zakleszczeń może dochodzić wtedy, kiedy w systemie zachodzą jednocześnie cztery warunki:

- ✖ wzajemne wykluczanie (zasoby są niepodzielne)
- ✖ przetrzymywanie i czekanie (procesy przetrzymują zasoby w czasie oczekiwania na inne)
- ✖ brak wywłaszczeń (zasobów nie można zawłaszczać, dopóki są użytkowane)
- ✖ czekanie cykliczne (istnieje łańcuch cykliczny procesów takich, że następny zamawia zasoby zajęte przez poprzednika)

GRAF PRZYDZIAŁU ZASOBÓW



(C) IISI d.KIK PCz 2013

Systemy operacyjne

7

GRAF PRZYDZIAŁU ZASOBÓW

Jeśli graf przydziału zasobów :

- ✖ nie zawiera cykli, to w systemie nie ma zakleszczonej procesów
- ✖ zawiera cykl, to doszło do zakleszczenia

Jeśli istnieje po kilka egzemplarzy zasobu każdego typu, to cykl w grafie jest warunkiem koniecznym, lecz nie wystarczającym do istnienia zakleszczenia.

(C) IISI d.KIK PCz 2013

Systemy operacyjne

8

STRATEGIE POSTĘPOWANIA Z ZAKLESZCZENIEM

- ✖ zapobieganie zakleszczeniom przez niespełnienie jednego z warunków
- ✖ wykrywanie i usuwanie zakleszczeń w oparciu o graf stanu
- ✖ unikanie zakleszczeń przez wykonywanie czynności wyprzedzających
- ✖ można też zlekceważyć problem zupełnie, uważając, że do zakleszczenia nigdy nie dojdzie

(C) IISI d.KIK PCz 2013

Systemy operacyjne

9

ZAPOBIEGANIE ZAKLESZCZENIOM

przez niespełnienie jednego z warunków:

- ✖ zmiana zasobu niepodzielonego w podzielny np. przez zastosowanie spoolera'
- ✖ procesy zamawiają wszystkie zasoby na początku i wtedy rozpoczynają działanie
- ✖ jeśli proces nie otrzyma zasobów zwrota już przydzielone i zamawia ponownie
- ✖ ustalony porządek zamówień uniemożliwia powstanie łańcucha cyklicznego

(C) IISI d.KIK PCz 2013

Systemy operacyjne

10

WYKRYWANIE I USUWANIE ZAKLESZCZEŃ W OPARCIU O GRAF STANU

- ✖ usunięcie wszystkich procesów uczestniczących w zakleszczeniu
- ✖ wznowienie procesów od punktu kontrolnego, jeśli istnieje
- ✖ usuwanie kolejnych procesów, aż do zaniku zakleszczenia
- ✖ kolejne zwalaszczanie zasobów

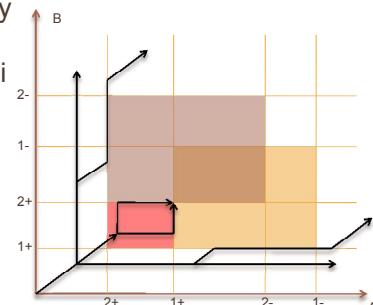
(C) IISI d.KIK PCz 2013

Systemy operacyjne

11

UNIKANIE ZAKLESZCZEŃ PRZEZ WYKONYWANIE CZYNNOŚCI WYPRZEDZAJĄCYCH

sprawdzanie, czy po przydiale zasobów nastąpi blokada



(C) IISI d.KIK PCz 2013

Systemy operacyjne

12

UNIKANIE ZAKLESZCZEŃ PRZEZ WYKONYWANIE CZYNNOŚCI WYPRZEDZAJĄCYCH

Algorytm bankiera

- ✗ roszczenie to maksymalna wielkość zamawianych zasobów w czasie trwania procesu
- ✗ zamówienie nie może przekroczyć roszczenia
- ✗ zamówienie jest przyjęte tylko wówczas, gdy po jego realizacji istnieje taki ciąg procesów, że wszystkie procesy dobiegną do końca

PLANOWANIE

Planowaniem nazywa się ustalanie, kiedy można wprowadzać nowe procesy do systemu i w jakiej kolejności je wykonywać.

Planowaniem zajmuje się proces planisty (scheduler) nazywany również planistą wysokiego poziomu, aby odróżnić go od planisty niskiego poziomu – dyspozytora.

DYSPOZYTOR

Decyzje o przydiale procesora mogą zapadać w następujących czterech sytuacjach:

- ✗ proces przeszedł od stanu aktywności do stanu czekania
- ✗ proces przeszedł od stanu aktywności do stanu gotowości
- ✗ proces przeszedł od stanu czekania do stanu gotowości
- ✗ proces kończy działanie.

PLANISTA - ZADANIA

- ✗ wprowadzanie nowych procesów
 - ✗ z puli prac (wsadowych)
 - ✗ uruchamianych przez użytkowników
- ✗ wyznaczanie priorytetów procesów
 - ✗ porządkowanie kolejki procesora
- ✗ implementacja przydziału zasobów
 - ✗ unikanie zakleszczeń
 - ✗ zapewnienie zrównoważenia systemu

PLANISTA

jest wywoływany, gdy:

- ✗ pojawiło się zamówienie na zasoby
- ✗ zasoby zostały zwolnione
- ✗ proces zakończył działanie
- ✗ do puli dodano nową pracę/użytkownik uruchomił nowy proces

KRYTERIA OCENY ALGORYTMÓW PLANOWANIA

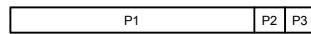
- ✗ Wykorzystanie procesora
 - ✗ powinno być od 40% do 90%
- ✗ Przepustowość
 - ✗ ilość procesów kończonych w jednostce czasu
- ✗ Czas cyklu przetwarzania
- ✗ Czas oczekiwania
 - ✗ w kolejce procesów gotowych
- ✗ Czas odpowiedzi
 - ✗ w systemach interaktywnych

ALGORYTMY PLANOWANIA

- ✗ Algorytm FCFS (First Come, First Served)
 - ✗ pierwszy zgłoszony - pierwszy obsłużony
- ✗ Algorytm SJF (Shortest Job First)
 - ✗ najpierw najkrótsza praca (zadanie)
- ✗ Algorytm SRTF (Shortest Remaining Time First)
 - ✗ najpierw najkrótszy pozostały czas / najpierw najkrótsza praca z zawieszaniem
- ✗ Algorytm priorytetowy
- ✗ Algorytm rotacyjny RR (Round-robin)
- ✗ Planowanie wielopoziomowe

ALGORYTM FCFS

- ✗ Dla procesów P1, P2, P3 o czasie trwania faz 24, 3 i 3:
- ✗ jeśli procesy nadejdą w porządku P1, P2, P3 i zostaną obsłużone w porządku FCFS:



- ✗ średni czas oczekiwania wynosi = 17 ms.
- ✗ Gdyby procesy nadeszły w kolejności P2, P3, P1 rezultat przedstawałby się jak na poniższym diagramie:



- ✗ średni czas czekania wynosiłby = 3 ms.

ALGORYTM SJF

- ✗ Algorytm najpierw najkrótsze zadanie pozbawiony jest wady FCFS, gdyż przydziela procesor procesowi posiadającemu najkrótszą następną fazę.
- ✗ Bieżący proces jest wywieszany przez nowy proces którego następna faza procesora jest krótsza od pozostałej części fazy procesu aktualnego.
- ✗ Algorytm ten daje minimalny średni czas oczekiwania.
- ✗ Długość faz procesora jest przewidywana na podstawie jego wcześniejszych faz, nie można jednoznacznie określić jaką długość będzie miała kolejna faza.

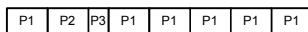


ALGORYTM PRIORYTETOWY

- ✗ W algorytmie tym każdemu procesowi przydziela się jakiś priorytet, liczbę całkowitą.
- ✗ Procesor zostaje przydzielony procesowi o najwyższym priorytecie.
- ✗ Algorytm może być niewywieszający albo wywieszający.
- ✗ Może wystąpić głodzenie procesów, polegające na tym, że procesy o niskim priorytecie mogą nigdy nie zostać wykonane. Aby temu zapobiec wprowadza się postarzanie, czyli stopniowe podwyższanie priorytetów procesów dłużej oczekujących.

ALGORYTM ROTACYJNY

- ✗ W algorytm ten wyposażono w wywłaszczenie i kolejkę cykliczną.
- ✗ Każdemu procesowi przydziela się mały odcinek czasu procesora. Proces wykonywany jest po upływie tego czasu wywłaszczański i przenoszony na koniec kolejki procesów gotowych.
- ✗ Dla n procesów w kolejce i kwantu czasu q, każdy proces dostaje $1/n$ czasu procesora, których wartość nie przekracza q. Czas oczekiwania procesu na dostęp do procesora nigdy nie przekracza $(n-1)*q$ jednostek czasu.



PLANOWANIE WIELOPOZIOMOWE

- Algorytm ten rozdziela kolejkę procesów gotowych na osobne kolejki. Stosuje się go gdy możliwe jest łatwe zaliczenie procesów do kilku różnych grup:
- ✗ pierwszoplanowe, drugoplanowe/wsadowe
 - ✗ systemowe, interakcyjne, redagowania interakcyjnego, wsadowe, studenckie.
 - ✗ Poszczególne kolejki mogą używać różnych algorytmów planowania.

PLANOWANIE WIELOPOZIOMOWE ZE SPRZĘŻENIEM ZWROTNYM

- ✗ W tym algorytmie możliwe jest przenoszenie procesów między kolejkami.
- ✗ Proces, który wykorzystuje cały przydzielony czas zostaje przeniesiony do kolejki o niższym priorytecie.
- ✗ Proces szybko oddający procesor jest przenoszony do kolejki o wyższym priorytecie.

OCENA ALGORYTMÓW

Istnieje wiele algorytmów planowania, z których każdy ma swoiste parametry.

Aby móc dokonywać wyboru algorytmu można wziąć pod uwagę kilka miar, takich jak np.:

- ✗ maksymalizacja wykorzystania procesora
- ✗ maksymalizacja przepustowości

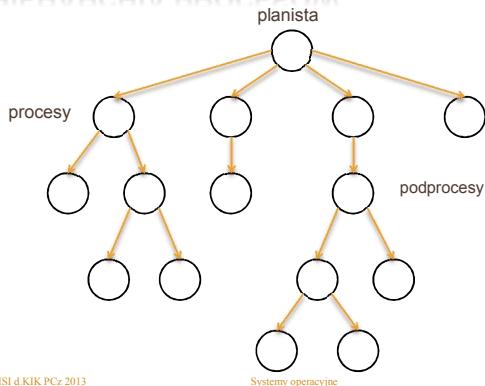
KRYTERIA PODJMOWANIA DECYZJI PRZEZ PLANISTĘ

- ✗ procesom mającym wiele zasobów można nadać wysoki priorytet
- ✗ procesom mającym wiele zasobów przydziela się kolejne
- ✗ przy przydzielaniu pamięci należy uwzględnić zasady zbioru roboczego
- ✗ procesom systemowym nadaje się priorytet zależny od ilości zadań jakie wykonują
- ✗ procesom obsługi urządzeń nadaje się wysoki priorytet
- ✗ jeśli nie zapobiega się zakleszczeniom powinno się ich unikać
- ✗ koszt planowania nie powinien przekraczać osiąganych zysków

HIERARCHIA PROCESÓW

- ✗ najczęściej:
planista - system op. - procesy – podprocesy
- ✗ planista jest przodkiem wszystkich procesów
- ✗ procesy mogą:
 - ✗ tworzyć podprocesy
 - ✗ przydzielać podprocesom część własnych zasobów
 - ✗ nadawać priorytet własnym podprocesom

HIERARCHIA PROCESÓW



(C) IISI d.KIK PCz 2013

3

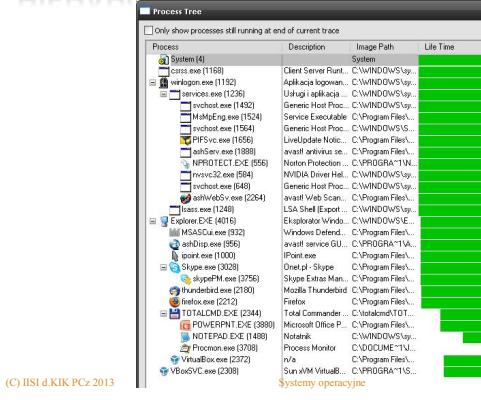
HIERARCHIA PROCESÓW W LINUKSIE

```
int i=0;BoxClient->~"["(BoxClient)~"
|acpid
|avahi-daemon
|bluetooth-kit-dae--62"["[console-kit-dae]
|crond
|dhclient
|gconf-2
|hald-hal-runner + hald-addin-acpi
|hald-addin-input
|lsm
|2"["[hald-addin-store]
|lplugd
|lpd
|lpd
|kdenlive + draknapsashot-ap
|l-klauner
|l-karneval + kwin
|l-konsole + [konserver]
|l-net-applet
|l-python
|l-systray
|lxappearance-kde
|kdm -X
|kdm -startkde -kwrapred
|krofty
|krofty
|krofty + bash -su - bash -ptree
|krofty - "[(konsole]
|korgac
|mplayer - pickup
|mpg3
|d'mingity]
|plasma - 2"["[plasma]
|profiler
|pulseaudio + gconf-helper
|pulseaudio - "[(pulseaudio]
|syslogd
|syslogd - 2"["[splashify]]
|syslogd
|udev
|vboxadditions-timesyn
```

Systemy operacyjne

32

HIERARCHIA PROCESÓW W WINDOWS



KONTROLA ZASOBÓW I ROZLICZENIA

Instalacje:

✗ usługowe

x rozliczanie

✗ wewnętrzne

✗ ograniczanie możliwości

racjonowanie zasobów

- ✗ krótkoterminowe – maksymalne bieżące
 - ✗ długoterminowe – użycie zasobów w ciągu tygodnia/miesiąca