1. Różnica między systemem operacyjnym, a systemem komputerowym.

System operacyjny to program komputerowy, gwarantujący prawidłową i wydajną pracę komputera i stowarzyszonych zasobów (peryferia, pliki, oprogramowanie). Udostępnia programom i użytkownikom usługi do realizacji zadań programistycznych. Musi być stabilny. Powinien decydować w jakiej kolejności będą dostępne zasoby. Jest dystrybutorem zasobów i arbitrem w sytuacjach konfliktowych. Jest programem sterującym nadzorującym działanie programów.

System komputerowy to jednostka centralna (CPU) komunikująca się z urządzeniami poprzez sterowniki urządzeń, połączonych wspólną szyną, z dostępem do wspólnej pamięci (PAO). Każdy sterownik urządzenia odpowiada za określony typ urządzenia. W skład systemu komputerowego wchodzą: CPU, PAO, pamięci masowe, urządzenia dodatkowe (skanery, steamery, drukarki, tunery TV, karty dźwiękowe).

2. Wyjaśnij na czum polega buforowanie, spooling, wieloprogramowość i wielozadaniowość.

Wielozadaniowość stanowi rozszerzenie wieloprogramowość. Procesor wykonuje na przemian wiele różnych zadań, przy czym przełączania następują tak często, że użytkownicy mogą współdziałać z każdym programem podczas jego działania.

Wieloprogramowość sprawia, że system operacyjny decyduje za użytkownika. Wieloprogramowość jest najważniejszym aspektem planowania zadań. Jeden użytkownik nie zdoła utrzymać cały czas w aktywności procesora lub urządzeń I/O. Wieloprogramowość zwiększa wykorzystanie procesora gdyż tak organizuje zadania, aby procesor miał zawsze któreś z nich do wykonania.

Buforowanie jest to przechowywanie w pamięci danych, które są przesyłane między dwoma urządzeniami lub między urządzeniem a aplikacją. Bufor jest to obszar pamięci, w którym przechowywane są te dane.

Spooling to jednoczesna, bezpośrednia praca urządzeń. Spooling używa dysku jako bufora do czytania z urządzeń We oraz do przechowywania plików Wy do czasu, aż urządzenia wyjściowe będą w stanie je przyjąć. Spooling umożliwia zdalne przetwarzanie danych. Monitor pobiera/wysyła dane łączem komunikacyjnym z własną szybkością bez udziału procesora, który jest tylko informowany o zakończeniu.

3. Co to jest system rozproszony.

System rozproszony w ostatnich latach pojawiła się tendencja do rozdzielania obliczeń między wiele procesorów. Procesory mają własną pamięć lokalną i komunikują się na przykład za pomocą szybkich szyn danych lub linii telefonicznych. Systemy takie nazywa się rozproszonymi. Celowość budowy: podział zasobów, przyspieszenie obliczeń, niezawodność, komunikacja.

System czasu rzeczywistego stosowany jest, gdy istnieją ostre wymagania na czas wykonania operacji lub przepływu danych, ma ściśle zdefiniowane stałe ograniczenia czasowe. Przetwarzanie danych musi się zakończyć przed upływem określonego czasu, w przeciwnym razie system nie spełnia wymagań. W systemach RTS rzadko spotyka się pamięć wirtualną.

4. Opisz sposoby komunikacji pomiędzy urządzeniami I/O.

Urządzenia I/O komunikują się ze sobą za pomocą przerwań. Urządzenie zgłasza przerwanie i chęć dostępu do innego urządzenia. Procesor ustawia zawartość rejestrów w sterowniku urządzenia. Sterownik sprawdza dane w tych rejestrach i określa typ działania. Cała operacja zachodzi przy udziale CPU.

5. Tablica Stanu Urządzeń. Przykład tablicy.

Tablica Stanów Urządzeń (Device Status Table), której elementy odnoszą się do poszczególnych urządzeń. Tablica zawiera typ urządzenia, jego adres i stan (odłączone, bezczynne, zajęte).Jeżeli urządzenie jest zajęte z powodu przyjęcia zamówienia, to odpowiadający mu element tablicy zawiera rodzaj zamówienia i inne parametry. Jeśli urządzenie We/Wy wymaga obsługi to wysyła przerwanie: Po wystąpieniu przerwania SO określa, które urządzenie spowodowało przerwanie. Pobiera z tablicy urządzeń informacje o stanie danego urządzenia i zmienia je, odnotowując wystąpienie przerwania. Zakończenie operacji urządzenia We/Wy również jest sygnalizowane przerwaniem. Jeśli są następne zamówienia oczekujące na dane urządzenie, to SO rozpoczyna ich realizacje. Na koniec procedura obsługi przerwania urządzenia We/Wy zwraca sterowanie.

6. Co oznacza pojęcie dualny tryb pracy systemu operacyjnego.

Sprzęt rozróżnia dwa oddzielne tryby pracy: tryb użytkownika, tryb monitora (tryb systemu). Tryb monitora jest nadrzędny. W sprzęcie istnieje bit trybu, którego stan wskazuje bieżący tryb pracy (0: monitor ||| 1: Użytkownik). Został on wprowadzony, by system odróżniał działania wykonywane na zamówienie SO od działań na zamówienie użytkownika (np. rozruch systemu przebiega w trybie monitora).

7. Co to są rozkazy uprzywilejowane.

Rozkazy uprzywilejowane to potencjalnie niebezpieczne rozkazy kodu maszynowego. Ich wydzielenie daje dalszą ochronę. Sprzęt pozwala wykonywać rozkazy uprzywilejowane tylko w trybie monitora. W trybie użytkownika wywołanie takiego rozkazu – spowoduje awaryjne przekazanie kontroli do SO. Przykładowe rozkazy: włączenie i wyłączenie systemu przerwań, modyfikacja rejestrów zarządzania pamięcią lub rejestrów czasomierza, rozkaz HALT.

8. W jakim celu wykorzystujemy rejestr bazowy i graniczny.

Rejestr bazowy i graniczny wykorzystujemy w celu ochrony pamięci. W rejestrze bazowym przechowywany jest najmniejszy dopuszczalny adres fizyczny pamięci. Rejestr graniczny przechowuje rozmiar obszaru pamięci. Ochronę realizuje sprzęt CPU przez porównywanie każdego adresu powstałego w trybie pracy użytkownika w wyżej wymienionych rejestrów.

BAZA ← ADRES ← BAZA + REJESTR GRANICZNY → OK.

9. Jakie znasz rodzaje systemów wieloprocesorowych i na czym polegają różnice w ich działaniu.

Systemy wieloprocesorowe: pewna liczba procesorów ściśle współpracuje ze sobą, dzieląc szynę komputera, zegar, a czasami pamięć i urządzenia zewnętrzne. Systemy takie nazywa się ściśle powiązanymi. Pozwala to na zwiększenie przepustowości. Systemy wieloprocesorowe dają możliwość wspólnego użytkowania urządzeń I/O. Mamy systemy z wielo-powtarzaniem symetrycznym (Na każdym procesorze działa identyczna kopia systemu operacyjnego. Kopie komunikują się ze sobą w zależności od potrzeb. Procesory

mogą korzystać ze wspólnych struktur danych) i asymetrycznym (Każdy procesor ma przypisane zadanie. Procesor główny planuje prace procesorom podporządkowanym. Inne procesory czekają na instrukcje od procesora głównego, albo zajmują się swoimi uprzednio określonymi zadaniami).Można też wyróżnić systemy tolerujące awarie - dwa procesory: podstawowy i zapasowy, z podstawowego dane kopiowane na zapasowy, w wyniku uszkodzenia procesora podstawowego uruchamiany jest zapasowy.

10. Opisz składowe systemu operacyjnego.

Proces: program wykonywalny – zadanie wsadowe, program użytkownika, buforowanie wyjścia. Proces korzysta z zasobów: czas CPU, pamięć, pliki, urządzenia I/O. Proces otrzymuje zasoby podczas jego tworzenia lub są one przydzielane podczas jego działania.

Pamięć operacyjna to magazyn wspólnie eksploatowany przez CPU i urządzenia I/O. PAO jest jedyną pamięcią jaką CPU może adresować bezpośrednio. Wykonywany program w PAO jest zaadresowany za pomocą adresów bezwzględnych. Program zakończy działanie, a jego miejsce w pamięci jest oznaczone jako wolne.

Pliki to informacja przechowywana na nośnikach różniących się budową i organizacją fizyczną. SO odwzorowuje pliki na fizyczne nośniki i umożliwia do nich dostęp za pomocą urządzeń pamięci.

Pamięć zewnętrzna: często używana. Programowe komponenty przechowywane są na dysku.

Urządzenia I/O: system operacyjny powinien ukrywać przed użytkownikiem szczegóły specyfiki sprzętowych urządzeń I/O. Służy do tego podsystem I/O (zarządzanie pamięcią, ogólny interfejs do modułów sterujących urządzeń, moduły sterujące urządzeń sprzętowych).

System ochrony: mechanizm nadzorujący dostęp programów, procesów, użytkowników do zasobów komputerowych. Zawiera sposoby określania, co i jak ma podlegać ochronie.

Interpreter poleceń: interfejs między użytkownikiem a systemem operacyjnym. Część systemów operacyjnych zawiera interpreter poleceń w swoim jądrze. W MS-DOS i UNIX interpreter poleceń jest specjalnym programem wykonywanym przy rozpoczynaniu zadania lub gdy użytkownik rejestruje się w systemie.

Usługi systemu operacyjnego: SO dostarcza usług programom i użytkownikom tych programów (wykonanie programu, operacje I/O, komunikacja, wykrywanie błędów, przydzielanie zasobów, ochrona).

11. Opisz co to są funkcje systemowe i wymień sposoby przekazywania do nich parametrów.

Funkcje systemowe tworzą interfejs między wykonywanym programem a SO. Standardowo można z nich korzystać za pomocą rozkazów w języku asemblera. Można wywoływać funkcje systemowe w niektórych językach wyższego poziomu, poprzez wywołanie procedury, która podczas wykonywania programu wykonuje funkcje systemowe, albo też funkcje systemowe dołączone są bezpośrednio do wykonywanego programu.

Metody przekazywania parametrów: umieszczenie parametrów w rejestrach CPU, blok lub tablica pamięci – adres bloku przekazuje się jako parametr za pośrednictwem rejestru, parametry składuje się na stosie za pomocą programu skąd będą zdejmowane.

12. Przedstawić sposoby działania interpretera poleceń.

Interpreter poleceń to interfejs pomiędzy SO a użytkownikiem , może być rozwiązany na dwa sposoby: Interpreter ma zaimplementowane w sobie obsługi każdego polecenia. Wywołanie polecenia powoduje skok do części interpretera wywołującej odpowiednie funkcje systemowe dla tego polecenia. Interpreter nie ma zaimplementowanej obsługi poleceń. Polecenia przechowywane są w oddzielnych plikach. Wpisanie komendy powoduje załadowanie odpowiedniego pliku z poleceniem i wykonanie go.

13. Co to jest maszyna wirtualna i problemy z jej realizacją.

Maszyna wirtualna jest procesem użytkowym, ale musi mieć szansę być monitorem. Sensowne połączenie sprzętu i funkcji tworzy maszynę wirtualną. Maszyna wirtualna nie dostarcza dodatkowych funkcji, tworzy jedynie interfejs identyczny z podstawowym sprzętem. Każdy proces otrzymuje (wirtualną) kopię komputera będącego podstawą systemu. Użytkownicy otrzymują własne maszyny wirtualne i mogą traktować je jak maszynę bazową. Fizyczna maszyna bazowa ma dwa tryby pracy: użytkownika i monitora.

Korzyści:

- W środowisku VM istnieje pełna ochrona różnorodnych zasobów systemowych.
- Każda maszyna wirtualna jest całkowicie odizolowana od innych maszyn wirtualnych.

Problemy:

- Nie ma bezpośredniej możliwości wspólnego użytkowania zasobów.
- Pewnym problemem maszyny wirtualnej jest realizacja systemu dysków.

14. Co to jest proces oraz struktura bloku kontrolnego procesora.

Proces jest to program w trakcie wykonywania, jednostka pracy w systemie z podziałem czasu. Proces to nie tylko kod programu. To wartość licznika rozkazów i rejestrów procesora, stos procesu przechowujący parametry procedur, to adresy powrotne i zmienne tymczasowe oraz sekcja danych zawierająca zmienne globalne. Dwa procesy związane z jednym programem będą traktowane jako dwie oddzielne sekwencje wykonania.

Blok kontrolny procesu (Process Control Block - PCB) reprezentuje proces i zawiera:
○ dane identyfikujące proces; identyfikatory: procesu, procesu rodzicielskiego; użytkownika.
○ stan procesu; licznik rozkazów -adres następnego rozkazu do wykonania w procesie; rejestry procesora.
○ informacje o planowaniu przydziału procesora -priorytet procesu, wskaźniki do kolejek porządkujących zamówienia, inne parametry planowania.
○ informacje o zarządzaniu pamięcią -zawartości rejestrów granicznych, tablice stron.
○ informacje do rozliczeń.
○ ilość zużytego czasu procesora i czasu rzeczywistego, ograniczenia czasowe, numery kont, numery procesów itp..
○ informacje o stanie We/Wy -wykaz otwartych plików itd. Informacje o stanie rejestrów i licznika rozkazów są przechowywane podczas przerwań, aby proces mógł być później kontynuowany.

15. Scharakteryzuj stany w jakich może znaleźć się proces.

- Nowy (proces został utworzony).
- Aktywny (są wykonywane instrukcje).
- Oczekiwanie (czeka na wystąpienie zdarzenia).
- Gotowy (czeka na przydział procesora).
- Zakończony (zakończył działanie).

15a. Model dwustanowy i pięciostanowy procesu.

Dwustanowy model procesu: Proces może być wykonywany przez procesor lub nie. Dopuszczalne są dwa stany procesu: „AKTYWNY" lub „nie Aktywny". Proces „nie Aktywny" jest widziany przez SO i oczekuje na uruchomienie w kolejce procesów oczekujących. Co pewien czas proces przetwarzany przez CPU zostaje przerwany i SO wybiera, za pomocą dyspozytora, następny proces do uruchomienia. Zatrzymany proces przechodzi w stan „nie Aktywny" i trafia do kolejki a proces wybrany w stan „AKTYWNY".

Pięciostanowy model procesu: Rozwiązaniem problemu modelu dwustanowego jest podział niedziałających procesów: Gotowe do pracy (Czekające), Zablokowane.

W każdej chwili tylko jeden proces może być aktywny na określonym procesorze, ale wiele procesów może być gotowych do działania lub czekających.

15b. Rola PCB w przełączaniu procesów. Przedstawienie za pomocą rysunku.

1. Zachować kontekst procesora (licznik rozkazów, inne rejestry, itp.).
2. Zaktualizować blok sterowania aktualnie wykonywanego procesu (zmiana stanu procesu, nadanie stosownych wartości innym polom np.: o przyczynie przerwania procesu, dane do rozliczeń).
3.Przenieść blok sterowania procesem do odpowiedniej kolejki (Gotowy, Zablokowany//Czekający).
4. Wybrać do przetwarzania inny proces.
5. Zaktualizować blok sterowania nowo wybranego procesu (zmienić stan procesu na Działający).
6. Zaktualizować struktury danych zarządzające pamięcią.
7. Odtworzyć kontekst procesora do stanu, jaki miał miejsce, gdy uruchamiany proces został ostatnio zatrzymany.

15c. Przedstaw i omów diagram planowania procesów z uwzględnieniem kolejki procesów i kolejki urządzeń We/Wy.

Kolejka jest listą wiązaną: nagłówek zawiera wskaźniki do pierwszego i ostatniego bloku kontrolnego procesu na liście. Każdy PCB ma pole wskazujące następną pozycję w kolejce procesów gotowych. Każde urządzenie ma własną kolejkę.

Proces który otrzymał procesor może:

1. zamówić operację We/Wy, wskutek czego trafia do kolejki procesów oczekujących na We/Wy.
2. utworzyć nowy pod proces i oczekiwać na jego zakończenie.
3. być przymusowo usunięty (przerwanie) i przeniesiony znowu do kolejki procesów gotowych.

W przypadkach (1) i (2) proces zostanie w końcu przełączony ze stanu oczekiwania do stanu gotowości i przeniesiony do kolejki procesów gotowych.

16. Wyjaśnij pojęcia: proces niezależny, współpracujący, wątek.

Proces niezależny nie może oddziaływać na inne procesy wykonywane w systemie, a te nie mogą wpływać na jego działanie. Proces niezależny nie dzieli żadnych danych z innym procesem.

Proces współpracujący może wpływać na inne procesy w systemie lub inne procesy mogą oddziaływać na niego. Dzieli dane z innymi procesami.

Wątek lub lekki proces to podstawowa jednostka wykorzystania procesora o składzie: licznik rozkazów, zbiór rejestrów, obszar stosu. Wątek nie przełącza pamięci operacyjnej. Wątki poziomu użytkownika wykorzystują wywołania biblioteczne zamiast odwołań do systemu dzięki czemu przełączanie wątków nie wymaga wzywania SO i przerwań związanych z przechodzeniem do jego jądra.

16a. Różnica pomiędzy procesem a wątkiem.

Pojęcie proces i wątek odnoszą się do wykonywalnego kodu. Proces ma przestrzeń adresowa pamięci wirtualnej i informacje takie jak podstawowy priorytet. Proces ma jeden lub więcej wątków, które są jednostkami wykonywania zarządzanymi przez jadro. Każdy watek ma własny stan, priorytet, przypisanie do procesora i informacje rozliczeniowe. Każdy proces posiada własna przestrzeń adresowa, natomiast watki posiadają wspólna sekcje danych.

16b. Wątki poziomu jądra, wątki poziomu użytkownika.

Wątki poziomu użytkownika: rezygnują z zarządzania wykonywaniem przez jadro i robią to same.
Wątki poziomu jądra: watki poziomu jadra są często implementowane poprzez dołączanie do każdego procesu tabeli/listy jego wątków. W tym rozwiązaniu system zarządza każdym wątkiem wykorzystując kwant czasu przyznany dla jego procesu-rodzica.

17. Opisz rodzaje planistów oraz opisz na czym polega planowanie zadań przez każdego z nich.

Planista krótkoterminowy wybiera jeden proces spośród procesów gotowych do wykonania i przydziela mu procesor. Musi często wybierać nowy proces dla procesora. Proces może działać przez kilka milisekund, a potem przejść w stan oczekiwania wydawszy zamówienie na operację I/O. Musi być bardzo szybki.
Planista długoterminowy wybiera procesy z pamięci masowej i ładuje je do pamięci w celu wykonania. Działa rzadziej – między kolejnymi procesami mogą upływać minuty, nadzoruje stopień wieloprogramowości (liczbę procesów w pamięci), może być wywoływany gdy jakiś proces opuszcza system.

18. Przedstaw możliwe schematy rozwiązania problemu producent-konsument.

Proces producent wytwarza informacje, które zużywa proces konsument. Aby umożliwić działanie współbieżne, musimy dysponować buforem jednostek. Podczas gdy producent tworzy pewną jednostkę, konsument może zużywać inną. Procesy producenta i konsumenta muszą podlegać synchronizacji, aby konsument nie próbował konsumować tych jednostek, które nie zostały jeszcze wyprodukowane. W takiej sytuacji konsument musi czekać na wyprodukowanie tego, co chce konsumować.
Problem z nieograniczonym buforem (unbounded-buffer): nie ma ograniczeń na rozmiar bufora. Producent może produkować nieustannie, zaś konsument musi czekać na nowe jednostki.

Problem z ograniczonym buforem (bounded-buffer): zakłada się, ze bufor ma ustaloną długość. Konsument czeka, gdy bufor jest pusty, a producent czeka, jeśli bufor jest pełny.

19. Co to jest przełączanie kontekstu.

Przełączanie kontekstu (context switch) to przełączanie procesora od innego procesu z przechowaniem stanu starego procesu i załadowanie przechowanego stanu nowego procesu. Podczas przełączania systemu nie wykonuje żadnej użytecznej pracy. Przy dużej liczbie zbiorów rejestrów, przełączenie sprowadza się do zmiany wartości wskaźnika bieżącego zbioru rejestrów. Złożoność SO zwiększa nakład pracy podczas przełączania. Przygotowując pamięć dla następnego zadania, należy przechować przestrzeń adresową bieżącego procesu. Przełączanie kontekstu jest wąskim gardłem systemu operacyjnego.

20. Do czego służy dispatcher.

Ekspedytor (dispatcher) – odrębny moduł w planowaniu przydziału procesora, który przekazuje procesor do dyspozycji wybranego procesu przez planistę krótkoterminowego.

Obowiązki ekspedytora:
- przełączanie kontekstu.
- przełączanie do trybu użytkownika.
- wykonanie skoku do komórki w programie użytkownika w celu wznowienia programu.

Ekspedytor jest wywoływany podczas każdego przełączania procesu.

21. Jakie wielkości można brać pod uwagę przy planowaniu wykorzystania czasu procesora.

- Wykorzystanie procesora: w rzeczywistym systemie powinno mieścić się w przedziale (40-90)%
- Przepustowość (throughput): liczba procesów kończonych w jednostce czasu. Dla długich procesów 1 na godzinę, dla krótkich 10 procesów na sekundę.
- Czas cyklu przetwarzania: czas upływający między chwilą nadejścia procesu do systemu, a chwilą zakończenia procesu.
- Czas oczekiwania: suma czasów, spędzonych przez proces w kolejce procesów gotowych do działania.
- Czas odpowiedzi: czas upływający między przedłożeniem zamówienia a pojawieniem się odpowiedzi w systemach interakcyjnych, nie obejmuje czasu potrzebnego na wyprowadzenie tej odpowiedzi. Jest uzależniony od szybkości urządzeń We/Wy.

Wykorzystanie procesora i przepustowość muszą być maksymalne, zaś czasy minimalne.

22. Wymień algorytmy planowania przydziału czasu procesora. Uszeregowanie przykładowe 3 procesów.

Metoda FCFS (First-Come, Fisrt-Served): proces który pierwszy zamówi procesor, pierwszy go otrzyma.
Metoda SJF (Shortest-Job-First): proces mający najkrótsza następną fazę procesora otrzyma procesor pierwszy.
Planowanie priorytetowe: procesom przypisuje się priorytet i przydziela się procesor procesowi o najwyższym priorytecie.
Planowanie rotacyjne RR (Round-Robin): algorytm dla systemów z podziałem czasu.

22a. Przedstaw wywłaszczające planowanie przydziału czasu procesora dla SJF (dla 4 procesów).

Co stanie się, jeśli proces o wyższym priorytecie zechce przeczytać lub zmienić dane jądra w chwili, w której korzysta z nich inny proces o niższym priorytecie? Wysoko priorytetowy proces musiałby czekać na zakończenie procesu o niższym priorytecie. Sytuacja ta nosi nazwę odwrócenia priorytetu (priority - inversiori).Może powstać łańcuch procesów korzystających z zasobów potrzebnych procesowi wysoko priorytetowemu. Problem rozwiązuje protokół dziedziczenia priorytetów (priority-inheritance protocol), w którym wszystkie procesy (używające zasobów potrzebnych procesowi wysoko priorytetowemu) dziedziczą wysoki priorytet, dopóki nie przestaną korzystać ze spornych zasobów. Po skończeniu działania tych procesów ich priorytety wracają do swojej pierwotne wartości.

23. Zdefiniuj różnice pomiędzy planowaniem wywłaszczającym, a niewywłaszczającym. Zaklasyfikuj znane Ci algorytmy planowania do którejś z wymienionych kategorii.

Zmiana przydziału procesora może nastąpić gdy proces:
1. Przeszedł od stanu Aktywności do stanu Czekania (zamówienie na We/Wy lub rozpoczęcie czekania na zakończenie działania procesu potomnego).
2. Kończy działanie.
3. Przeszedł od stanu Aktywności do Gotowości (np. wskutek wystąpienia przerwania).4.przeszedł od stanu Czekania do Gotowości (np. po zakończeniu operacji We/Wy).

Sytuacja 1 i 2 nie daje wyboru. Procesor otrzyma nowy proces z kolejki procesów gotowych.
Planowanie nie wywłaszczeniowe: proces otrzymując procesor, zachowuje go do czasu swojego zakończenia lub przejścia do stanu Czekania. Np. Metoda FCFS.
W sytuacjach innych niż 1 i 2 algorytm planowania jest typu wywłaszczeniowego. Np. Planowanie Rotacyjne RR.
Algorytmy, które mogą być i wywłaszczające i niewywłaszczające: SJF, Planowanie priorytetowe.

24. Planowanie priorytetowe z wykorzystaniem kolejek oraz kolejki ze sprzężeniem zwrotnym.

Planowanie Priorytetowe

Procesom przypisuje się priorytet i przydziela się procesorowi o najwyższym priorytecie. Procesy o równych priorytetach planuje się metodą FCFS. Algorytm SJF jest szczególnym przypadkiem planowania priorytetowego. Priorytety należą do pewnego, ustalonego przedziału liczb całkowitych. Przyjmujemy, że im mniejsza liczba tym wyższy priorytet.

Planowanie rotacyjne RR

Algorytm dla systemów z podziałem czasu. Jest podobny do FCFS, z tym że w celu przełączania procesów dodano do niego wywłaszczanie. Kolejka procesów gotowych do wykonania jest traktowana jak kolejka cykliczna. Planowanie wielopoziomowych kolejek ze sprzężeniem zwrotnym Istnienie osobnych kolejek dla procesów pierwszoplanowych i drugoplanowych wyklucza możliwość przechodzenia procesu z jednej

kolejki do drugiej. Planowanie wielopoziomowych kolejek ze sprzężeniem zwrotnym umożliwia przemieszczanie procesów między kolejkami. Ideą jest rozdzielenie procesów o różnych długościach faz procesora.

25. Wyjaśnij w jakim stopniu poniższe algorytmy planowania mogą faworyzować krótkie procesy: FCFS, rotacyjny, wielopoziomowe kolejki ze sprzężeniem zwrotnym.

FCFS – pierwszy zgłoszony, pierwszy obsłużony. Proces, który pierwszy zamówił procesor, pierwszy go otrzyma. Blok kontrolny jest dołączany na koniec kolejki. Wolny proces przydziela się procesorowi z czoła kolejki. Jeśli przychodzą trzy procesy w chwili t=0 d długościach P1=6ms, P2=3ms, P3=2ms. Jeśli będą obsługiwane w kolejności P1, P2, P3, to średni czas oczekiwania wyniesie (0+6+9)/3=5ms. Natomiast jeśli będą obsłużone w kolejności P3, P2, P1, to średni czas oczekiwania wyniesie (0+2+5)/3=2.33ms. Krótsze procesy mogą być faworyzowane ze względu na krótszy średni czas oczekiwania. Planowanie rotacyjne to algorytm dla systemów z podziałem czasu. Ustala się jednostkę czasu zwaną kwantem czasu (zwykle od 10 do 100ms). Kolejka procesów gotowych do wykonania jest traktowana jak kolejka cykliczna. Planista przydziału procesora przegląda kolejkę i każdemu procesowi przydziela odcinek czasu nie dłuższy niż jeden kwant czasu. Średni czas oczekiwania w metodzie rotacyjnej nie może być długi. Planowanie wielopoziomowych kolejek ze sprzężeniem zwrotnym umożliwia przemieszczanie procesów między kolejkami. Proces zużywający dużo czasu zostanie przeniesiony do kolejki o niższym priorytecie. Ustawia to procesy ograniczone przez I/O i procesy interakcyjne w kolejkach o wyższych priorytetach.

26. Koncepcja algorytmu rozwiązującego problem Czytelników i Pisarzy.

Pierwszy problem czytelników i pisarzy: zakłada się że żaden czytelnik nie powinien czekać, chyba że właśnie pisarz otrzymał pozwolenie na używanie obiektu dzielonego. Żaden czytelnik nie powinien czekać na zakończenie pracy innych czytelników tylko z tego powodu, że czeka pisarz.

Druga wersja: Zakłada się, że jeśli pisarz jest gotowy, to rozpoczyna wykonanie swojej pracy tak wcześnie, jak to tylko możliwe. Jeśli pisarz czeka na dostęp do obiektu, to nowy czytelnik nie rozpocznie czytania. Obie wersje mogą powodować głodzenie.

26a. Koncepcja algorytmu rozwiązującego problem 5 filozofów.

Pięciu filozofów spędza życie na myśleniu i jedzeniu. Filozofowie siedzą przy okrągłym stole. Na środku stołu stoi miska ryżu, a naokoło leży pięć pałeczek. Kiedy filozof myśli, to tylko myśli. Czasami filozof jest głodny. Wówczas próbuje ująć dwie pałeczki lezące najbliżej jego miejsca. Za każdym razem filozof może podnieść tylko jedną pałeczkę. Kiedy filozof zdobędzie obie pałeczki, rozpoczyna jedzenie, nie rozstając się z pałeczkami ani na chwilę. Po spożyciu posiłku filozof odkłada obie pałeczki na stół i ponownie zatapia się w rozmyślaniach. Problem filozofów to klasyczne zagadnienie synchronizacji, odzwierciedlające konieczność przydzielania wielu zasobów do wielu procesów w sposób gorący zakleszczeniami i głodzeniem. Proste rozwiązanie zakłada, ze pałeczka jest semaforem. Semafor paleczka[5] = {1, 1, 1, 1, 1}. Proste rozwiązanie zapewnia, ze zadni dwaj sąsiedzi nie będą jedli jednocześnie, nie wyklucza powstania zakleszczenia. Rozwiązanie wolne od zakleszczeń nie eliminuje automatycznie możliwości blokowania nieskończonego. Semafor kierownik = 4 .Dodatkowy semafor kierownik o wartości początkowej 4, dopuszcza do rywalizacji o pałeczki co najwyżej czterech filozofów, gwarantując uniknięcie zakleszczenia.

27. Podaj mechanizm konwersji przestrzeni adresowych.

Wirtualna przestrzeń adresowa (to nie fizyczna) to tylko zakres adresów pamięci. Aby z niej skorzystać (bez wywołania błędu dostępu), trzeba najpierw przypisać do fragmentu pamięci wirtualnej pamięć fizyczna czyli dokonać mapowania. Program ma dostęp tylko do wirtualnej pamięci, natomiast system mapuje adresy wirtualne w fizyczne (nie muszą one stanowić ciągłego bloku). Gdy proces ładowany jest po raz pierwszy, względne odwołania do pamięci w jego kodzie zastępowane są odwołaniami do adresów absolutnych w PAO, określonych przez adres bazowy ładowanego procesu. Jednostka zarządzania pamięcią ***(Memory Management Unit - MMU)*** to urządzenie sprzętowe odwzorowujące adresy wirtualne na fizyczne w czasie działania programu. Program użytkownika działa na adresach logicznych, nigdy na rzeczywistych adresach fizycznych. Sprzęt odwzorowujący pamięć zamienia adresy logiczne na adresy fizyczne. Logiczna przestrzeń adresowa powiązana z odrębna, fizyczna przestrzenią adresowa jest podstawa zarządzania pamięcią.

28. Wyjaśnij różnicę między adresami logicznymi i fizycznymi.

Adres logiczny jest to adres wytworzony przez procesor (zwany adresem wirtualnym).

Adres fizyczny to adres umieszczony w rejestrze adresowym PAO.

Adresy logiczne i fizyczne są takie same podczas kompilacji oraz ładowania ale różne podczas wykonywania rozkazów.

29. Opisz strategie wyboru wolnego obszaru pamięci ze zbioru dostępnych dziur. Przydziel pamięć dla zadanych procesów i wskaż najlepszy algorytm.

Pierwsze dopasowanie: przydziela się pierwszą dziurę o wystarczającej wielkości. Szukanie rozpoczyna się od początku wykazu dziur lub od miejsca, w którym je ostatnio zakończono. Szukanie kończy napotkanie dostatecznie dużej dziury.

Najlepsze dopasowanie: przydziela się najmniejszą z dostatecznie dużych dziur. Należy przejrzeć całą listę, chyba że jest ona uporządkowana według wymiarów. Strategia zapewnia najmniejsze pozostałości po przydziale.

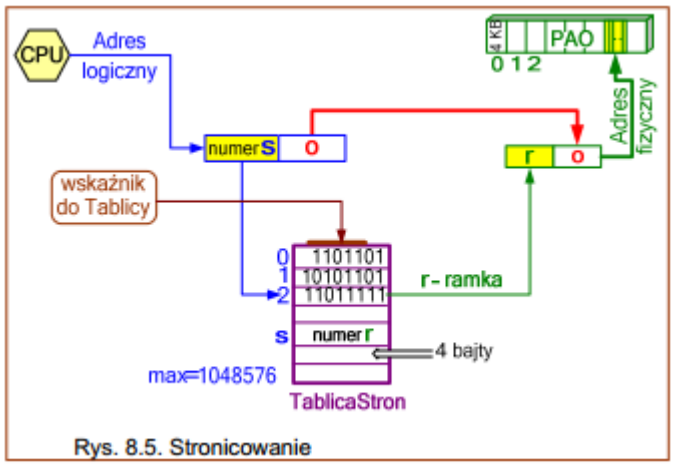
Najgorsze dopasowanie: przydziela się największą dziurę. Należy przeszukać całą listę, gdy nie jest uporządkowana według wymiarów. Po przydziale pozostaje największą dziura, która może okazać się bardziej użyteczna niż pozostałość wynikająca ze strategii najlepszego dopasowania.

Strategie wyboru pierwszej lub najlepiej dopasowanej dziury są lepsze od wyboru największej dziury zarówno pod względem zmniejszania czasu, jak i zużycia pamięci. Przydziały pamięci na bazie najlepszego dopasowania są z reguły szybsze.

30. Zastosowanie jednostki zarządzania pamięcią.

Jednostka zarządzania pamięcią (Memory Management Unit – MMU) to urządzenie systemowe odwzorowujące adresy wirtualne na fizyczne w czasie działania programu.

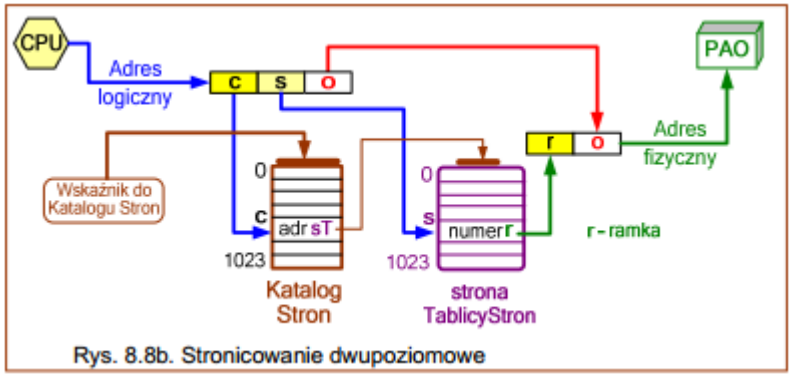
31. Stronicowanie. Opis oraz rysunek.



Rys. 8.5. Stronicowanie

Stronicowanie dopuszcza istnienie nieciągłości logicznej przestrzeni adresowej procesu – zezwala na przydział dowolnie dostępnych miejsc w pamięci fizycznej. Pamięć logiczna dzieli się na bloki tego samego rozmiaru zwane stronami. Pamięć fizyczna PAO dzieli się na bloki o stałej długości zwane ramkami. Stronicowanie eliminuje zewnętrzną fragmentację. Stronicowanie to podział pamięci logicznej na bloki tego samego rozmiaru. Stronicowanie rozdziela pamięć oglądaną przez użytkownika od pamięci fizycznej.

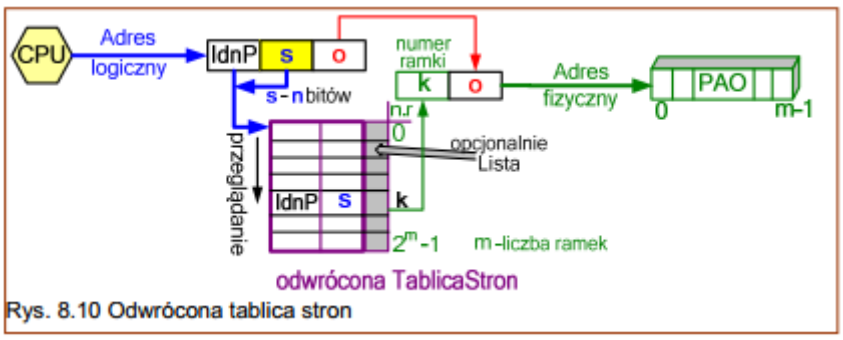
31a. Stronicowanie wielopoziomowe. Opis oraz rysunek.



Rys. 8.8b. Stronicowanie dwupoziomowe

Współczesne komputery umożliwiają stosowanie dużych przestrzeni adresów logicznych. W takim środowisko sama Tablica Stron zajmuje zbyt dużo ciągłego obszaru PAO. Niech system o 32-bitowej logicznej przestrzeni adresowej ma stronę o rozmiarze 4 KB. Tablica Stron może zawierać milion pozycji. Pozycja w tablicy ma 4B, stad każdy proces może potrzebować do 4 MB PAO na sama Tablice Stron. Zatem podział Tablicy Stron na części. 32-bitowy adres logiczny dzieli się na 20-bitowy numer strony i 12-bitowa odległość na stronie. Każda z dwu tablic zawiera 210 elementów, zaś ka(di element to 4 bajty, zatem każda z tablic także Katalog Stron maja wielkość 4 kB, co dokładnie odpowiada jednej ramce pamięci fizycznej. Negatywny wpływ stronicowania wielopoziomowego na wydajność systemu Przekształcenie adresu logicznego na fizyczny może wymagać 4-ch dostępów do pamięci, co zwiększa łączny czas realizacje jednego dostępu do PAO 5-ciokrotnie. Poprawę przynosi zastosowanie szybkiej pamięci podręcznej. Dla współczynnika trafień 98%: efektywny czas dostępu = 0,98 x 120 + 0,02 x 520 =128 ns. Przy dostępie do PAO 100ns, dodatkowe poziomy tablic wydłużają efektywny czas dostępu o 28%.

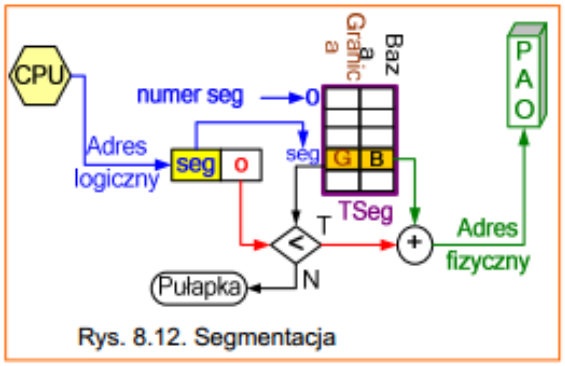
31b. Opisz zastosowanie mechanizmu Odwróconej Tablicy Stron. Rysunek.



Rys. 8.10 Odwrócona tablica stron

Odwrócona Tablica Stron ma po jednej pozycji dla każdej rzeczywistej ramki pamięci. Każda pozycja zawiera:
-informacje o procesie, do którego strona należny,
-adres logiczny strony przechowywanej w ramce pamięci rzeczywistej.
Wpis w Odwróconej Tablicy Stron to: < Identyfikator_procesu, numer_strony>
Struktura adresu logicznego: < IdnProcesu, numer_strony, odległość >
W systemie istnieje tylko jedna Tablica Stron. Odwołanie do pamięci, przekazuje cześć adresu logicznego: <IdnProcesu, numer_strony> do podsystemu pamięci. Następuje przeszukanie Odwróconej Tablicy Stron w celu dopasowania adresu. Jeśli dopasowanie powiedzie się (spełni r-ty element), to tworzy się adres fizyczny: <r, odległość>. Opisany schemat zmniejsza rozmiar pamięci potrzebnej do pamiętania wszystkich tablic stron. Zwiększa czas potrzebny do przeszukania tablicy przy odwołaniu do strony. Odwrócona Tablica Stron jest uporządkowana według adresów fizycznych. Przeglądanie dotyczy adresów wirtualnych, zatem należy przeszukać ja w całości. Czas przeszukiwania poprawiają rejestry pamięci asocjacyjnej, w których przechowuje się ostatnio zlokalizowane wpisy. Rejestry przeglądane są przed zaglądaniem do tablicy.

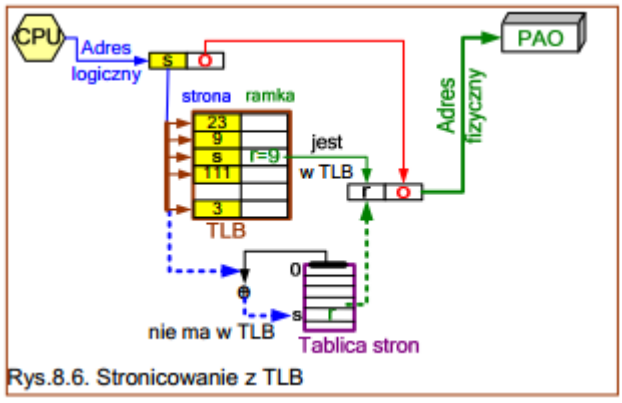
32. Segmentacja. Opis i Rysunek.



Rys. 8.12. Segmentacja

Segmentacja to schemat zarządzania pamięcią, w którym przestrzeń adresów logicznych jest zbiorem segmentów. Każdy segment ma nazwę i długość. Stronicowanie: użytkownik określa tylko pojedynczy adres, dzielony następnie przez sprzęt na numer strony i odległość – w sposób niewidoczny dla programisty. Adres logiczny tworzy para: <numer_segmentu, odległość>.

32a. Zastosowanie rejestru asocjacyjnego. Opis i rysunek.



Rys.8.6. Stronicowanie z TLB

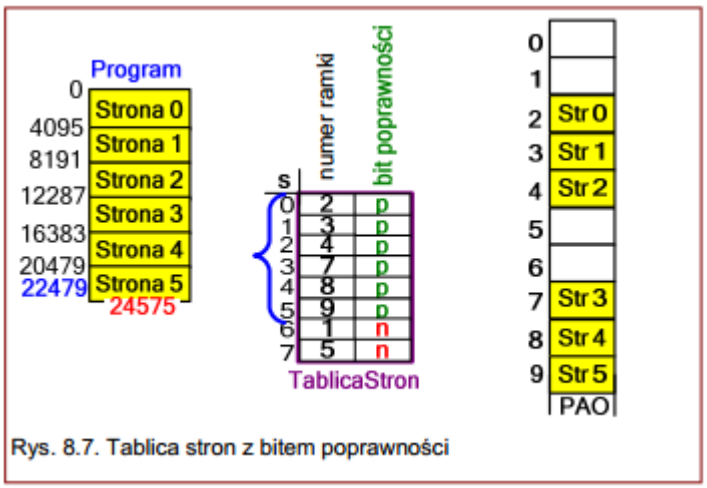
Rejestr asocjacyjny (associative registers) -szybka sprzętowa pamięć podręczna, zwana te (Buforem ***Translacji Adresów Stron (Translation Look-aside Buffers – TLB)***. Każdy rejestr asocjacyjny składa się z dwóch części: klucza i wartości. Porównanie danego obiektu z kluczami w TLB odbywa się równocześnie dla wszystkich kluczy. Zgodność obiektu z kluczem udostępnia pole wartości. Rejestr asocjacyjny zawiera kilka wpisów z Tablicy Stron. TLB zawiera numery stron i odpowiadające im ramki. Numer strony adresu logicznego porównywany jest ze zbiorem rejestrów asocjacyjnych. Jeśli numer strony zostanie odnaleziony to dostęp do numeru ramki jest natychmiastowy dostęp do PAO. Jeśli numeru strony nie ma w TLB , to trzeba odwołać się do miejsca w PAO, w którym przechowywana jest Tablica Stron. Dodatkowo dołącza się poszukiwany numer strony i ramki do rejestrów

asocjacyjnych, aby przy następnym odwołaniu numery te były łatwo osiągalne. Jeśli rejestry asocjacyjne są pełne, to SO wybiera któryś z nich do zastąpienia wartości. Po każdym wyborze nowej Tablicy Stron (np. każdorazowo po przełączaniu kontekstu), rejestry asocjacyjne zostają opróżnione. Inaczej w TLB zostałyby stare wpisy zawierające poprawne adresy wirtualne, lecz z błędnymi adresami fizycznymi, pozostałymi po poprzednim procesie.

33. Sposoby implementacji tablicy stron.

Tablica stron przydzielana jest do każdego procesu w większości systemów. Tablica stron może być zbiorem rejestrów o dużej szybkości działania. Ekspedytor określa zawartość tych rejestrów. Rejestrowa tablica stron jest wystarczająca dla małej tablicy stron (256 pozycji). Dużą tablicę stron przechowuje się w pamięci operacyjnej, a do wskazywania jej położenia służy rejestr bazowy tablicy stron.

33a. Tablica Stron z bitem poprawności – rysunek.



Rys. 8.7. Tablica stron z bitem poprawności

Każdy wpis w tablicy stron jest uzupełniany o dodatkowy bit poprawności (valid-invalid bit).Stan „poprawny" oznacza, ze strona, z która jest on związany, znajduje się w logicznej przestrzeni adresowej procesu strona jest dozwolona. Stan „niepoprawny" oznaczy, ze strona nie należy do logicznej przestrzeni adresowej procesu. Niedozwolone adresy są wychwytywane za pomocą sprawdzania stanu bitu poprawności. SO nadaje wartość bitowi poprawności w odniesieniu do każdej strony, zezwalając lub zakazując na korzystanie ze strony.

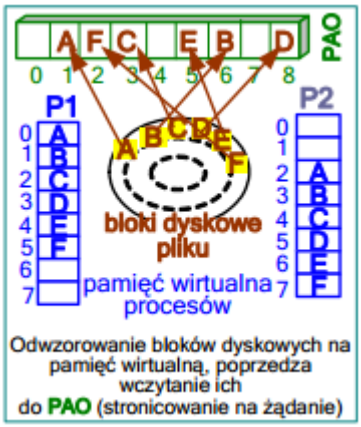
34. Co to jest fragmentacja pamięci i jakie znasz jej rodzaje? Wyjaśnij występujące różnice.

Fragmentacja – przeszukiwanie zbioru dziur różnych wielkości w chwili zamówienia przez procesor pamięci w celu znalezienia dziury wystarczająco dużej dla danego procesu.
Zewnętrzna fragmentacja – pamięć jest poszatkowana na duża liczbę małych dziur, suma wolnych obszarów w pamięci wystarcza na spełnienie zamówienia, al. Nie tworzą one spójnego obszaru.
Wewnętrzna fragmentacja – różnica miedzy wielkością pamięci zamawianej przez procesor a przydzielonej procesowi. Zazwyczaj przyłącza się dziury będące różnicą (a to bardzo małe dziury) do większych przydziałów. Należy minimalizować niepełne wykorzystanie pamięci w ramach partycji np. Przypisując procesowi najmniejsza partycje, w której się jeszcze zmieści.
Upakowanie: takie przemieszczanie zawartości pamięci, aby cała wolna pamięć znalazła się w jednym bloku. Upakowanie nie zawsze jest możliwe.

35. Opisz na czym polega segmentacja stronicowana.

Stronicowanie i segmentacja maja zalety i wady. Można łączyć oba schematy w celu wykorzystania zalet każdego z nich. Systemie stronicowania/segmentacji przestrzeń adresowa użytkownika jest podzielona na szereg segmentów, a każdy z nich ulega dalej podziałowi na strony o ustalonej wielkości, równej wielkości ramki pamięci głównej. Jeżeli segment jest mniejszy niż strona, to zajmuję jedną stronę.

36. Na czym polega stronicowanie na żądanie. Rysunek.



Proces stanowi ciąg stron, nie jest wielką i ciągłą przestrzenią adresową. Procedura leniwej wymiany: nie dokonuje się wymiany strony w pamięci jeśli nie jest to konieczne. Procedura stronicująca zajmuje się stronami procesu. Zgaduje, które strony będą w użyciu przed ponownym wysłaniem ich na dysk. Nie wymienia całego procesu, lecz sprowadza do pamięci tylko strony niezbędne. Niezbędne są środki sprzętowe do odróżnienia stron pozostających w PAO od stron na dysku. Oznaczenie strony jako niepoprawnej nie wywołuje żadnych skutków, jeśli proces nigdy się do niej nie odwoła.

37. Kiedy występują błędy strony? Opisz działania podejmowane przesz system operacyjne, gdy wystąpi błąd braku strony.

Błędy strony mogą wystąpić np. gdy strona nie należy do logicznej przestrzeni adresowej procesu, strona jest dozwolona, lecz aktualnie znajduje się na dysku.

Procedura obsługo błędu braku strony:
1. Sprawdzić w wewnętrznej tablicy, czy odwołanie do pamięci było dozwolone.
2. Jeśli odwołanie było niedozwolone to proces zostaje zakończony.
3. Jeśli było dozwolone, lecz zabrakło właściwej strony w pamięci, to sprawdzić tę stronę.
   a. wyszukać wolną ramkę.
   b. zamówić pobranie z dysku potrzebnej strony.
   c. po zakończeniu pobrania z dysku, zmodyfikować tablicę wewnętrzną procesu oraz tablicę stron
   d. wznowić wykonanie przerwanego rozkazu

Po wystąpieniu braku strony można wznowić proces dokładnie w tym samym miejscu i stanie- jeżeli żądana strona jest już dostępna - dzięki przechowaniu stanu przerwanego procesu.

38. Współczynnik trafienia, efektywny czas dostępu do pamięci – przykład obliczeniowy.

Współczynnik trafień (hit ratio): procent numerów stron odnajdywanych w rejestrach asocjacyjnych. Niech współczynnik trafień wynosi 80%. Niech przeglądniecie rejestrów asocjacyjnych zabiera 20ns, a dostęp do pamięci 100ns, i numer strony jest w rejestrach, wówczas odwzorowywany dostęp do PAO zajmuje 120 ns. Jeśli nie powiedzie się odnalezienie numeru strony w rejestrach asocjacyjnych (20 ns), to należy sięgnąć do pamięci po Tablice Stron i numer ramki (100 ns), po czym odwołać się do właściwego słowa w pamięci (100 ns) co łącznie daje 220 ns. Efektywny czas dostępu do pamięci (effective memory-access time) wymaga zastosowania do każdego z dwóch w/wym. przypadków wagi wynikającej z prawdopodobieństwa ich wystąpienia. efektywny czas dostępu = 0.80 x 120 + 0.20 x 220 = 140 ns Dla współczynnika trafień 98% mamy: efektywny czas dostępu = 0,98 x 120 + 0,02 x 220 = 122 ns. Współczynnik trafień zależny jest od liczby rejestrów asocjacyjnych.

39. Scharakteryzuj algorytmy zastępowania stron + przykład + ciąg odniesień + przykład.

Zastępowanie stron **–** powiększanie stopnia wieloprogramowości prowadzi do nad przydziału pamięci. Wykonuje się 6 procesów, każdy ma rozmiar 10 stron i faktycznie używa 5 stron. Fizycznie jest to 40 ramek.
Algorytm: Powinien minimalizować częstość braków stron. Algorytm ocenia się na podstawie wykonania go na pewnym ciągu odniesień do pamięci i zsumowania liczby braków stron.
Ciąg odniesień **–** to zapis adresu każdego odwołania do pamięci na podstawie śledzenia danego systemu. Liczba danych ulega redukcji po uwzględnieniu:
- dla znanego rozmiaru stron można brać numery stron a nie adres
- odwołanie się do strony s, następujące bezpośrednio po odwołaniu do strony s, nigdy nie powoduje braku strony.

Aby określić liczbę brakujących stron dla ciągu odniesień należy znać liczbę dostępnych ramek.
Algorytm FIFO kojarzy z każda stroną czas wprowadzenia jej do pamięci. Do zastąpienia strony wybiera się stronę najstarszą. Nie jest niezbędne odnotowywanie czasu sprowadzenia strony. Kolejka FIFO może przechowywać wszystkie strony przebywające w pamięci. Do zastąpienia będzie delegowana strona z czoła kolejki. Strona wprowadzana do pamięci zostanie ulokowana na końcu kolejki. Algorytm LRU zastępowania najdawniej używanych stron (Least Recently Used - LRU).Zastąpiona zostaje strona, która nie była używana od najdłuższego czasu. Algorytm LRU kojarzy z każdą stroną czas jej ostatniego użycia. Algorytm LRU jest wolny od anomalii Belady'ego. Algorytm optymalny OPT lub MIN Zastąp tę stronę, która najdłużej nie będzie (przyszłość) używana. Algorytm optymalny jest wolny od anomalii Belady'ego. Algorytmu optymalny gwarantuje najmniejszą z możliwych częstość braków stron dla danej liczby ramek.

39b. Problemy z implementacją algorytmu LRU.

Problemem implementacji algorytmu LRU jest określenie porządku ramek na podstawie czasu ich ostatniego użycia. W praktyce stosuje się dwie metody implementacji: Licznik i Stos. Licznik: Do każdej pozycji w Tablicy Stron dołączamy rejestr czasu użycia. Wykorzystuje się zegar logiczny lub licznik. Wskazania zegara zwiększane są z każdym odniesieniem do pamięci. Gdy występuje odniesienie do pamięci, zawartość rejestru zegara jest kopiowana do rejestru czasu użycia należącego do danej strony w Tablicy Stron. Dla każdej strony dysponujemy „czasem" ostatniego do niej odniesienia. Zastępujemy stronę z najmniejszą wartością czasu. Schemat wymaga przeglądania Tablicy Stron w celu znalezienia strony najdawniej używanej, jak również zapisywania w pamięci pola czasu użycia w Tablicy Stron. Rejestry czasu użycia wymagają uaktualnienia również przy wymianach Tablic Stron(powodowanych planowaniem przydziału procesora).Może powstać nadmiar w rejestrze zegara. Stos: Utrzymywany jest stos numer stron. Przy każdym odwołaniu do strony jej numer wyjmuje się ze stosu i umieszcza na jego szczycie. Na szczycie stosu jest strona ostatnio użyta, na spodzie jest strony najdawniej użyta. Trzeba wydobyć pozycje z wnętrza stosu stosuje się listę dwukierunkową ze wskaźnikami do czoła i końca listy. Wyjęcie i ulokowanie strony na szczycie stosu wymaga maksymalnie zmiany 6-ciu wskaźników. Każde uaktualnienie listy jest czasochłonne. Do zastąpienia strony nie jest potrzebne przeszukiwanie listy, ponieważ wskaźnik końcowy listy, wskazujący na dno stosu, identyfikuje najdawniej używana stronę.

40. Co to jest anomalia Belady'ego i gdzie ona występuje. Kiedy (gdzie) nie występuje.

Występuje w algorytmie FIFO natomiast algorytmy LRU oraz OPT (MIN) są wolne od tej anomalii. W niektórych algorytmach zastępowanie stron współczynnik braków stron może wzrastać ze wzrostem wolnych ramek. Dobrze to ilustruje ciąg odniesień: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5. W algorytmie FIFO liczba braków stron dla 4-ch ramek wynosi 10, zaś dla 3-ch wynosi 9.

41a. Algorytmy przybliżone zastępowania stron.

Algorytm dodatkowych bitów odwołań: Dla każdej strony przeznacza się bajt odwołań (b bitów) w tablicy pamięci operacyjnej) W regularnych okresach przerwanie zegarowe przekazuje sterowanie do SO. Dla każdej strony SO wprowadza bit odniesienia na najbardziej znaczącą pozycje bajta odwołań, przesuwając pozostałe bity o jedną pozycję w prawo, z utratą bitu na pozycji najmniej znaczącej. Interpretując 8-bitowe słowa jako liczby bez znak, za stronę najdawniej używaną i kandydującą do zastąpienia przyjmuje się stronę, której odpowiada najmniejsza liczba. Niech S1 = 00011100, S2 = 00011101 to strona S1 była używana później niż strona S2.

Algorytm drugiej szansy: Podstawą algorytmu zastępowania na zasadzie drugiej szansy jest algorytm FIFO. Po wybraniu strony sprawdza się dodatkowo bit odniesienia. Gdy bit odniesienia =1 dana strona dostaje drugą szansę i jej bit odniesienia jest zerowany, zaś czas jej przybycia ustawia się na czas bieżący, a pod uwagę bierze się następną stronę wynikającą z porządku FIFO. Strona, której dano drugą szansę, nie będzie uwzględniona przy zastępowaniu dopóki nie zostaną zastąpione inne strony (lub otrzymują drugą szansę) Implementacja algorytmu jest w postaci kolejki cyklicznej.

42. Na czym polega szamotanie systemu operacyjnego i w jaki sposób można temu zapobiec.

Jeśli proces nie będzie miał nominalnej liczby ramek to szybko wystąpi brak strony. Wtedy któraś ze stron musi być zastąpiona. Wszystkie strony są aktywnie używane, trzeba zastąpić stronę, która ze chwilkę będzie potrzebna. W procesie będą następowały kolejno po sobie braki stron. Proces wymienia jakąś stronę, po czym sprowadza ja z powrotem. Taka aktywność to szamotanie. Proces szamoce się, jeśli spędza więcej czasu na stronicowaniu niż na wykonywaniu. Zapobiec szamotaniu można dostarczając procesowi tyle ramek, ile potrzebuje.

43. Co to jest model zbioru roboczego?

Opiera się na założeniu, ze program ma charakterystykę strefowa. Parametr A definiuje okno zbioru roboczego. Istota modelu jest sprawdzanie ostatnich A odniesień do stron. Zbiór roboczy to zbiór stron, do których nastąpiło ostatnich A odniesień. Jeśli strona jest aktywnie używana to będzie znajdować się w zbiorze roboczym. Gdy strona przestanie być używana to wypadnie ze zbioru roboczego po A jednostkach czasu odliczonych od ostatniego do niej odwołania. Dokładność zbioru roboczego zależy od wyboru parametru A: Jeśli A będzie za mały to nie obejmie całego zbioru, jak będzie za duży to będzie zachodził na kilka stref programu. Najważniejsza cecha zbioru roboczego jest jego rozmiar: Ogólne zapotrzebowanie na ramki to: Z =a i RZR Jeśli łączne zapotrzebowanie jest większe niż ogólna liczba dostępnych ramek Z > m, to powstanie szamotanie, gdyż niektóre procesy nie otrzymują wystarczającej liczby ramek.

Zastosowanie modelu:
SO dogląda zbioru roboczego w każdym procesie i przydziela każdemu procesowi tyle ramek, ile wymaga rozmiar jego zbioru roboczego. Dodatkowe ramki zezwalają na rozpoczęcie nowego procesu. Gdy suma rozmiarów zbiorów roboczych wzrasta i zaczyna przekraczać łączną liczbę dostępnych ramek, wówczas SO wybiera proces, którego wykonanie trzeba będzie wstrzymać. Strony procesu są usuwane z pamięci, a jego ramki przydziela się innym procesom.

44. Porównaj przydział globalny i lokalny ramek pamięci.

Zastępowanie globalne **–** umożliwia procesom wybór ramki ze zbioru wszystkich ramek, nawet gdy ramka jest w danej chwili przydzielona do innego procesu – jeden proces może zabrać ramkę drugiemu procesowi. Proces o wyższym priorytecie może zwiększać liczbę swoich ramek kosztem procesu o niższym priorytecie.

Zastępowanie lokalne **–** ogranicza wybór do zbioru ramek przydzielonych do danego procesu.

W algorytmie globalnym proces może wybierać jedynie ramki przydzielone innym procesom, zwiększając liczbę przydzielonych mu ramek, ale nie może kontrolować własnej częstości występowania braków stron.

45. Wymień metody, które mogą zwiększać skuteczność stronicowania.

Stronicowanie wstępne (prepaging) może zapobiegać wysokiej aktywności stronicowania we wstępnej fazie procesu. Strategia zakłada jednorazowe wprowadzenie do pamięci wszystkich stron, o których wiadomo że będą potrzebne. Kolejną metodą zwiększającą skuteczność stronicowania może być wybór odpowiedniego rozmiaru strony.

Małe strony :
- dają lepsze wykorzystanie pamięci gdyż maleje fragmentacja wewnętrzna.
- można je lepiej zlokalizować co może zmniejszyć ilość przesyłanej informacji.
- dają się lepiej dopasować do sfer programu.
- dają lepszą rozdzielczość co pozwala na wydzielenie tylko tych fragmentów pamięci które są rzeczywiście potrzebne.
- mogą ograniczyć ilości informacji przekazywanej na We/Wy i zmniejszyć ogólny rozmiar przydzielanej pamięci.

Większe strony:
- każdy aktywny proces ma własną kopię tablicy stron.
- ułatwiają minimalizację czasu operacji We/Wy.
- pozwalają na minimalizacje braku stron.

46. Koszty i korzyści przy używaniu techniki pamięci wirtualnej.

Zalety:
1. możliwość wykonywania procesów chociaż nie są one w całości przechowywane w pamięci operacyjnej.
2. programy mogą być większe niż pamięć fizyczna.
3. pamięć wirtualna pozwala utworzyć abstrakcyjną pamięć główną w postaci olbrzymiej tablicy do magazynowania informacji i oddzielić pamięć logiczną, oglądaną przez użytkownika od pamięci fizycznej.
4. wykonuje się mniej operacji we- wy bo cały program jest ładowany do pamięci.***Wady:***
1. pamięć wirtualna może istotnie obniżyć wydajność programu w przypadku niestarannego użycia

47. Sposoby zarządzania wolnymi obszarami dyskowymi.

Wektor bitowy **–** lista wolnych obszarów można implementować w postaci wektora bitowego. Każdy blok jest reprezentowany przez 1 bit. Dla bloku wolnego bit ma wartość 1, dla bloku przydzielonego wartość 0. Metoda wydajnie odnajduje pierwszy wolny blok lub n kolejnych wolnych bloków na dysku. Zastosowanie wektorów bitowych jest mało wydajne jeśli nie można przechowywać całego wektora w PAO.
Lista powiązana – Można powiązać ze sobą wszystkie wolne bloki dyskowe i przechowywać wskaźnik do pierwszego wolnego bloku w specjalnym miejscu na dysku oraz pod ręcznie – w pamięci. Pierwszy blok zawiera wskaźnik do następnego wolnego bloku itd. Metoda nie jest wydajna – aby przejrzeć listę, należy odczytać każdy blok, co wymaga znaczniej ilości czasu na operacje We/Wy.
Grupowanie **–** można przechowywać adresy n wolnych bloków w pierwszym wolnym bloku. Pierwszych n-1 z tych bloków to bloki rzeczywiście wolne. Ostatni blok zawiera adresy następnych n wolnych bloków.
Zliczanie **–** kilka przyległych do siebie bloków można przydzielić i zwalniać jednocześnie, zwłaszcza wtedy, gdy stosuje się algorytm przydziału całkowitego.

Zamiast wykazu n wolnych adresów dyskowych można przechowywać:

- adres pierwszego wolnego bloku,

- liczbę n wolnych bloków następujących bezpośrednio po nim.

48. Metody przydziału miejsca na dysku:

a) Przydział ciągły **–** każdy plik musi zajmować ciąg kolejnych bloków na dysku. Adresy dyskowe definiują na dysku uporządkowanie liniowe. Jeśli z dyskiem kontaktuje się tylko jedno zadanie, dostęp do bloku b+1 po bloku b nie wymaga na ogół ruchu głowicy. Przy dostępie sekwencyjnym system plików pamięta adres dyskowy ostatniego bloku, do którego było odniesienie Bezpośredni dostęp do bloku k w pliku zaczynającym się od bloku b realizuje natychmiast operacja: b+k. Trudnością jest znalezienie miejsca na nowy plik. Strategie pierwszego dopasowania i najlepszego dopasowania są powszechnie stosowane do wybrania wolnego obszaru z listy dostępnych obszarów.

b) Przydział listowy **–** plik jest lista powiązanych ze sobą bloków dyskowych, znajdujących się gdziekolwiek na dysku. Katalog zawiera wskaźnik do 1-go i ostatniego bloku pliku. Każdy blok zawiera wskaźnik do nast. bloku. Wskaźniki te nie są dostępne dla użytkownika. Nie ma zewnętrznej fragmentacji. Nie trzeba znać rozmiaru pliku w chwili jego tworzenia.

c) Przydział indeksowy **–** W przydziale tym wskaźniki do bloków rozrzucone są wraz z blokami po całym dysku i mogą być osiągane po kolei – niewydajny dostęp bezpośredni. Przydział indeksowy skupia wskaźniki w jednym miejscu – w bloku indeksowym. Każdy plik ma własny blok indeksowy, będący Tablica Adresów bloków dyskowych. Pozycja k-ta w bloku wskazuje na k-ty blok pliku. Katalog zawiera adres bloku indeksowego. Wada przydziału indeksowego jest marnowanie przestrzeni. Wskaźniki bloku indeksowego zajmują na ogol więcej miejsca niż wskaźniki przy przydziale listowym. Plik ma tylko jeden lub dwa bloki. W przydziale listowym tracimy miejsce tylko dla jednego wskaźnika na blok, w przydziale indeksowym trzeba przydzielić cały blok nawet wówczas, gdy tylko jeden lub dwa wskaźniki będą w nim różne od NULL.

Różnice miedzy listowym i indeksowym:
Przydział indeksowy: Relatywnie mały, gdyż każdy plik musi mieć blok indeksowy. Duży blok indeksowy umożliwia pomieszczenie wszystkich wskaźników do wielkiego pliku. Wadą przydziału indeksowego jest marnowanie przestrzeni. Wskaźniki bloku indeksowego zajmują na ogół więcej miejsca niż wskaźniki przy

przydziale listowym.
Schemat listowy: blok indeksowy mieści się w jednym bloku dyskowym i można połączyć kilka bloków indeksowych. Blok indeksowy może zawierać mały nagłówek z nazwą pliku oraz zbiór stu pierwszych adresów bloków dyskowych. Następny adres (ostatnie słowo w bloku indeksowym) będzie miał wartość null (dla małego pliku) lub będzie wskaźnikiem do innego bloku indeksowego (dla dużego pliku).Realizacja dostępu bezpośredniego jest niewydajna. Potrzebna dodatkowa przestrzeń zajmowana przez wskaźniki. Pliki powiązane są wskaźnikami rozrzuconymi po całym dysku.

49. Metody planowania dostępu do dysku:

Napęd dyskowy jest jak jednowymiarowa tablica bloków logicznych, które sekwencyjnie odwzorowane są na fizyczne sektory dysku.
a) Algorytm FCFS: „pierwszy zgłoszony – pierwszy obsłużony”. Kolejka zaczyna się przesuwać przez kolejne zamówienia w kolejce, wada jest ze głowica gwałtownie się wychyla.
b) Algorytm SSTF: obsługuje się wszystkie zamówienia sąsiadujące z bieżąca pozycja głowicy zanim nastąpi jej przemieszczenie w dalsze rejony. Jest to „najpierw najkrótszy czas przeszukiwania”, czyli zamówienie z najmniejszym czasem przeszukiwania względem bieżącej pozycji głowicy.
c) Algorytm SCAN: głowica cały czas przeszukuje cały dysk od początku do końca. Ramie głowicy przesuwając się po dysku obsługuje wszystkie zamówienia po drodze.
d) Algorytm C-SCAN: Głowica przesuwa się od krawędzi dysku do środka, obsługuje napotkane po drodze zamówienia. Gdy osiągnie skrajne wychylenie, wraca do krawędzi zewnętrznej, NIE obsługuje zamówień w drodze powrotnej.
e) Algorytm LOOK i C-LOOK: Głowica przemieszcza się miedzy skrajnymi zamówieniami w każdym kierunku i nie dochodzi do skrajnego położenia na dysku. Te algorytmy są tak nazywane, bo głowica „patrzy” czy w danym kierunku są jeszcze jakieś zamówienia.

50. Organizacja struktury katalogowej. Wady i zalety.

a) System plików podzielony jest na strefy, nazwane tomami. Każdy tom zawiera co najmniej jedna strefę, mieszczącą pliki i katalogi. Mogą wystąpić strefy jako logiczne struktury kilku dysków.
b) Strefa zawiera informacje o zawartych w niej plikach. Informacje są przechowywane w pozycjach katalogu urządzenia, zawierającego dla każdego pliku danej strefy info takie jak: nazwa, położenie, rozmiar.

Katalog jednopoziomowy:
Wszystkie pliki są ujęte w tym samym katalogu. Łatwo utworzyć i obsługiwać. Pliki w tym samym katalogu, musza mieć jednoznaczne nazwy. Tworzenie plików o jednoznacznych nazwach staje się trudne przy wzroście liczby plików.
Katalog dwupoziomowy:
Rejestracja użytkownika w systemie uruchamia przeglądanie Głównego Katalogu Plików, który jest indeksowany nazwami użytkowników lub numerami kont. Każdy użytkownik ma własny katalog plików.
Katalog drzewiasty:
System plików MS-DOS ma strukturę drzewa. Pliki to liście, do których wiodą jednoznaczne nazwy ścieżek idące od korzenia poprzez podkatalogi. Umożliwia to użytkownikom dostęp do plików innych użytkowników. Struktura drzewiasta nie pozwala na dzielenie katalogów.
Acykliczny graf katalogów:
W grafie tym ten sam podkatalog lub plik może występować w dwu różnych katalogach. Pliki i podkatalogi dzielone mogą być implementowane na kilka sposobów. Jednemu plikowi może odpowiadać wiele

bezwzględnych nazw ścieżek. Do tego samego pliku mogą odnosić się różne nazwy plików.

Graf ogólny katalogów:
Należy zagwarantować, ze w acyklicznym grafie katalogów nie powstaną cykle. Algorytmy wykrywania cykli w grafach, wymagają dużych nakładów obliczeniowych, gdy graf znajduje się w pamięci dyskowej. Łączenie nieużytków w dyskowym systemie plików jest skrajnie czasochłonne.

51. Problemy występowania cykli w strukturach katalogowych.

Należy zagwarantować, ze w acyklicznym grafie katalogów nie powstaną cykle. Dowolne dodanie dowiązań do istniejącej drzewiastej struktury katalogowej, może przekształcić ją w ogólna strukturę grafu (wprowadzić cykle).Jeśli przejrzano podkatalog dzielony i nie znaleziono w nim poszukiwanego pliku, to należy unikać jego powtórzonego przeglądania. Wystąpienie cykli w istniejących katalogach, może prowadzić do powstania niekończącej się pętli przeszukiwania. Kiedy plik można usunąć? Gdy istnieją cykle, możliwe jest, ze licznik odniesień będzie rożny od zera nawet wtedy, gdy nie ma już odniesienia do katalogu lub pliku. Łączenie nieużytków (garbate collection) umożliwia ponowny przydział przestrzeni dyskowej i pozwala określić, kiedy usunięto ostatnie odniesienie. Następuje obchód systemu plików i oznaczenie wszystkiego, do czego można dotrzeć. Następnie wszystkie nie zaznaczone obszary zbiera się na wykazie wolnych przestrzeni. Łączenie nieużytków w dyskowym systemie plików jest skrajnie czasochłonne. Łączenie nieużytków jest konieczne tylko z powodu możliwości wystąpienia cykli w grafie. Trudnością jest unikanie cykli wówczas, gdy do struktury dodaje się nowe dowiązania. Algorytmy wykrywania cykli w grafach, wymagają dużych nakładów obliczeniowych, gdy graf znajduje się w pamięci dyskowej.

52. Wyjaśnić pojęcia: wieloprzetwarzanie symetryczne i asymetryczne.

Wieloprzetwarzanie symetryczne – (symetric multiprocessing), na każdym procesorze działa identyczna kopia systemu operacyjnego. Kopie komunikują się ze sobą w zależności od potrzeb. Należy zapewnić takie wykonanie operacji We/Wy, aby dane docierały do właściwych procesorów. Każdy procesor sam planuje swoje działanie, przegląda kolejkę procesów gotowych, z której wybiera proces do wykonania.

Wieloprzetwarzanie asymetryczne – (asymetric multiprocessing) każdy procesor ma przypisane inne zadanie. Procesor główny planuje i przydziela prace procesorom podporządkowanym. Inne procesy albo czekają na instrukcje od procesora głównego, albo zajmują się swoimi uprzednio określonymi zadaniami.

53. Wyjaśnij pojęcia: zgodność i spójność pamięci.

Część informacji katalogowych przechowywana jest w PAO i są na ogół nowsze niż odpowiadające im informacje na dysku, gdyż zapisanie na dysku informacji katalogowych zgromadzonych w pamięci podręcznej nie zawsze odbywa się wraz z ich uaktualnieniem.

Program sprawdzania ***spójności*** porównuje dane w strukturze katalogowej z blokami danych na dysku i próbuje usunąć wszelkie napotkane niezgodności. Jeśli stosuje się przydział listowy i istnieje połączenie miedzy każdymi dwoma blokami, to na podstawie bloków danych można zrekonstruować cały plik i odtworzyć strukturę katalogowa.

54. Przedstawić problemy efektywnego definiowania warstw systemu operacyjnego.

Problemem są wzajemne odwołania warstw: Obsługa pamięci dyskowej powinna być na niższym poziomie niż procedury obsługi pamięci gdyż korzystają one z pamięci dyskowej. Program obsługi pamięci dyskowej powinien być powyżej planowania przydziału procesora, gdyż program obsługi dysku może czekać na operacje we/wy a w tym czasie procesor może otrzymać nowe rozkazy. W dużych systemach program przydziału procesora może nie zmieścić procesów w pamięci operacyjnej, więc trzeba je zapisać w pamięci dyskowej, co powoduje potrzebę umieszczenia obsługi pamięci dyskowej poniżej planowania przydziału procesora.

55. Przedstawić problemy efektywnego definiowania warstw systemu operacyjnego.

Problemem są wzajemne odwołania warstw: Obsługa pamięci dyskowej powinna być na niższym poziomie niż procedury obsługi pamięci gdyż korzystają one z pamięci dyskowej. Program obsługi pamięci dyskowej powinien być powyżej planowania przydziału procesora, gdyż program obsługi dysku może czekać na operacje we/wy a w tym czasie procesor może otrzymać nowe rozkazy. W dużych systemach program przydziału procesora może nie zmieścić procesów w pamięci operacyjnej, więc trzeba je zapisać w pamięci dyskowej, co powoduje potrzebę umieszczenia obsługi pamięci dyskowej poniżej planowania przydziału procesora.

56. Przedstawić metody komunikacji między procesorowej.

Komunikacja między procesorowa: wspólne używanie Obiektów Jądra lub za pomocą przekazywania komunikatu.

57. Problemy planowania przydziału procesora w systemach czasu rzeczywistego.

Każda operacja powinna mieć gwarantowany maksymalny czas wykonania. Gwarancje nie są możliwe w systemie z pamięcią zewnętrzną.

1. System musi mieć planowanie priorytetowe. Priorytet nie może maleć w czasie.
2. Opóźnienie ekspediowania procesów do procesora musi być małe. (wpływają na to zwłaszcza powolne urządzenia (we/wy).

58. Na czym polega problem sekcji krytycznej.

System złożony z n procesów {P1,P2,….Pn} ma segment kodu, w którym procesy mogą zmieniać wspólne zmienne. Jeżeli jeden proces wykonuje swoja sekcje krytyczna, wówczas inny proces nie jest dopuszczony do wykonania swoje sekcji krytycznej. Wykonanie tej sekcji polega na wzajemnym wykluczaniu się w czasie. Należy udostępnić protokół, organizujący współprace procesów. Proces musi prosić o pozwolenie wejścia do swojej sekcji krytycznej. Kod realizujący taka prośbę to sekcja Wejściowa, po sekcji krytycznej następuje sekcja Wyjściowa, pozostały kod to Reszta.

Rozwiązanie problemu sekcji krytycznej:
a) Wzajemne wykluczanie – jeśli proces P1 działa w swojej sekcji to inny proces nie działa w sekcji krytycznej.

b) Postęp – tylko procesy nie wykluczające swoich reszt mogą kandydować do wejścia do sekcji krytycznej i wybór ten nie może być odwlekany w nieskończoność
c) Ograniczone czekanie – istnieje wartość graniczna liczby wejść innych procesów do ich sekcji krytycznej po tym, jak dany proces zgłosił chęć wejścia do swojej sekcji i zanim uzyskał pozwolenie.

Jeżeli dwa takie rozkazy wykonywane są współbieżnie, to wynik jest równoważny wykonaniu sekwencyjnemu w nieznanym porządku. Jeśli rozkazy Pobierz i Zapisz będą wykonane współbieżnie, to rozkaz Pobierz natrafi na starą albo nową wartość w pamięci, ale nie na jakąś kombinację ich obu.

59. Przedstawić algorytm obsługi sekcji krytycznej dla dwóch procesów.

Algorytm 1:
Pewnym komórkom pamięci nadajmy etykietę numer oraz statut zmiennej globalnej. Proces P0 lub P1 usiłujący wykonać swoja sekcje musi najpierw sprawdzić zawartość zmiennej numer. Jeśli jest ona równa numerowi procesu, wówczas może rozpoczynać inaczej musi czekać. Oczekujący proces systematycznie odczytuje wartość zmiennej numer do czasu aż uzyska zgodę na przetwarzanie sekcji krytycznej. Jest to aktywne oczekiwanie, gdyż wstrzymany proces nie robi nic produktywnego.

Wady:
- nie zachowuje wystarczającej informacji o stanie każdego procesu
- pamięta tylko któremu procesowi wolno wejść do sekcji krytycznej

Algorytm 2:
Potrzebne są informacje o stanie obu procesów, zatem oba procesy powinny mieć własne klucze do uruchamiania sekcji krytycznej, by w razie awarii jednego drugi mógł działać niezakłócony. Zmienna numer zastępuje dwuelementowa tablica Znacznik[] z wartościami początkowymi false. Proces P0, ustawia zmienna Znacznik[0]=true, co sygnalizuje gotowość wejścia procesu do sekcji krytycznej. Następnie P0 sprawdza czy P1 tez nie jest gotowy do wejścia do sekcji. Jeśli jest to P0 zaczeka, dopóki P1 nie zasygnalizuje, ze nie przebywa w sekcji krytycznej (Znacznik[1]=false). Wtedy P0 wchodzi do sekcji. Opuszczając sekcje krytyczna P0 nada swojemu znacznikowi wartość false, pozwalając drugiemu procesowi wejść do sekcji krytycznej.

Algorytm 3:
Należy umożliwić obserwowanie obu procesów oraz narzucić kolejność działania obu procesów. Procesy maja dwie wspólne zmienne: tablice Znacznik[0..1] oraz zmienna numer=(0,1). Na początku Znacznik[0..1]=false; Chcąc wejść do sekcji krytycznej, P0 najpierw ustawia Znacznik[0]=true, po czym zakłada ze również drugi proces chce wejść do sekcji. Gdy dwa procesy chcą wejść w tym samym czasie, to zmienna numer otrzyma wartość 0 oraz 1 w tym samych czasie. Przyjęta wartość zmiennej numer decyduje, który z procesów będzie miał prawo wejść do sekcji krytycznej w pierwszej kolejności. Algorytm jest poprawny, gdyż można dowieść, ze zapewnia wzajemne wykluczanie, spełnia warunek postępu i zachodzi warunek ograniczonego czekania.

60. Omówić problem sprzętowego wspomagania synchronizacji.

Sprzętowe wspomaganie synchronizacji polega na stosowania niepodzielnych rozkazów sprzętowych sprawdzających i zmieniających (systemy wieloprocesorowe), lub zakazywania przerwań podczas modyfikacji zmiennej dzielonej (system 1-procesorowy). Mechanizm rozkazowy sprowadza się do sprawdzania zmian pewnych zmiennych. Proces może się wykonać gdy zmienna ma odpowiednią wartość.

Rozpoczęcie wykonywania zmienia rozkazem tą zmienną na taką wartość, że żaden inny proces nie może się wykonywać do czasu gdy poprzedni nie zwolni sekcji krytycznej.

Zalety : Kontrola wielu sekcji krytycznych
Wady: aktywne czekanie, możliwość występowania zagłodzenia

61 – 62. Semafor klasyczny – implementacja i zastosowania + przykład.

Semafor jest zmienną niepodzielną dostępną za pomocą dwu niepodzielnych operacji czekaj i sygnalizuj. Semafory mogą wspomagać synchronizacje, eliminując problem wzajemnych wykluczeń. Implementacja: Jeśli wartość semafora jest dodatnia to proces może wejść do swojej sekcji krytycznej, kiedy wartość semafora jest niedodatnia to żaden proces nie może wejść do sekcji krytycznej. Inicjacja nadaje semaforowi wartość 1. Czyli proces może wejść do sekcji krytycznej. Wykonuje on też rozkaz czekaj który zmniejsza wartość semafora na 0. Żaden kolejny proces nie może wejść do sekcji krytycznej(0-niedodatnie). Zakończenie pierwszego procesu wywołuje rozkaz :Sygnalizuj i pierwszy z czekających procesów może przystąpić do realizacji sekcji krytycznej. Wymaga się aby Czekaj i Sygnalizuj były niepodzielne!

Przykład: algorytm 5 Filozofów (podany wyżej)

63. Semafor silny

W klasycznej definicji semafora z aktywnym czekaniem wartość semafora nigdy nie jest ujemna. W powyższej implementacji semafory mogą przyjmować wartości ujemne. Moduł ujemnej wartości semafora określa liczbę procesów czekających na ten semafor. Wynika to ze zmiany porządku instrukcji zmniejszania i sprawdzania wartości zmiennej w implementacji operacji Czekaj. Listę czekających procesów można zbudować, dodając pole dowiązań do bloku kontrolnego każdego procesu. Każdy semafor zawiera wartość całkowitą i wskaźnik do listy bloków kontrolnych procesów. Kolejka FIFO jest sposobem dodawania i usuwania procesów z Listy, gwarantującym ograniczone czekanie; semafor zawiera wskaźniki do początku i końca tej kolejki. Procesy zablokowane najdłużej opuszczają kolejkę pierwsze. Semafory implementujące tę zasadę nazywa się silnymi semaforami.

64. Funkcje związane z silnym semaforem.

Operacja BLOKUJ wstrzymuje proces, który ją wywołuje. Operacja OBUDZ(P) wznawia zablokowany proces P. Operacje te są dostarczane przez SO jako funkcje systemowe. Wznowienie procesu realizuje operacja BUDZENIE, zmieniająca stan procesu z CZEKANIA na GOTOWOŚĆ. Proces przechodzi do kolejki procesów Gotowych do wykonania.

65. Typy semaforów i ich charakterystyka.

Semafor binarny (binary semaphore) – jest zmienną logiczną i przybiera tylko wartości 0 lub 1. Semafor binarny nie pamięta liczby wykonanych na nim operacji podnoszenia. Semafor binarny rozwiązuje problem wzajemnego wykluczania. Semafor zliczający (counting semaphore) - przebiera dowolne wartości całkowite. Semafor silny i klasyczny omówione powyżej.

66. Co to są monitory i jaka jest ich funkcja.

Monitor to moduł programowy, składający się z jednej lub kilku funkcji,ciągu inicjującego i danych lokalnych. Monitory dostarczają mechanizmu dla programowania współbieżnego, skupiając

odpowiedzialność za poprawność w kilku modułach. Sekcje krytyczne, takie jak przydzielanie urządzeń We/Wy, pamięci, itd., skupione są w uprzywilejowanym programie. Można definiować oddzielne monitory dla każdego obiektu lub grupy obiektów. Monitor pełni rolę mechanizmu wzajemnych wykluczeń, jeśli dane w monitorze reprezentują zasoby, które powinny być chronione przed innymi procesami.

67. Omówić monitor ze zmiennymi warunkowymi.

Proces wywołuje monitor i pracując pod jego kontrolą może zostać w pewnym momencie zawieszony w oczekiwaniu na spełnienie określonego warunku. Potrzebny jest mechanizm, który: umożliwi zawieszenie procesu, zwolni monitor, aby inny proces mógł go wykorzystać. Gdy oczekiwany warunek wystąpi, a monitor znowu będzie dostępny, proces musi zostać wznowiony i uruchomić monitor od miejsca ostatniego zawieszenia. Zmienne warunkowe, dostępne tylko w obrębie MONITORA, stanowią dodatkowe narzędzia synchronizacyjne.

68. Omówić krótko problematykę transakcji niepodzielnych.

Transakcja to zbór instrukcji (operacji), które wykonują logicznie spójną całość. Jest ciągiem operacji czytaj i pisz zakończonych operacją zatwierdź lub Zaniechaj. Przetwarzanie transakcji wymaga zachowania ich niepodzielności pomimo awarii systemu. Zatwierdź oznacza że transakcja została przeprowadzona pomyślnie. Zaniechaj oznacza wystąpienie błędu logicznego. W takim przypadku Dane które zostały już zmienione przez transakcję muszą zostać przywrócone do stanu jaki miały przed wywołaniem transakcji. Transakcja zaniechana nie może pozostawić po sobie śladów w danych. Transakcja Zatwierdź nie może być cofnięta przez zaniechaj. Używa się rejestrów w pamięci stałej aby zapisywać dane na temat transakcji

69. Przedstawić schemat wzajemnych wykluczeń z udziałem monitorów.

Jeśli ten sam monitor wywoływany jest przez 2 procesy, to implementacja gwarantuje, że te procesy będą obsłużone w kolejności gwarantującej wzajemne wykluczanie. Składnia monitora hermetyzuje dane i działające na nich funkcjach w pojedyncze moduły. Interfejs monitora składa się ze zbioru funkcji, które operują na danych ukrytych w module. Monitor zapewnia wzajemne wykluczanie wykonań funkcji występujących w jego interfejsie.

70. Następuje awaria PAO. Jak zapewnić niepodzielność transakcji.

Odtwarzanie za pomocą rejestru. Niepodzielność można osiągnąć, zapisując w pamięci trwałej informacje określające wszystkie zmiany wykonywane przez transakcję w danych, do których ma ona dostęp. Metoda rejestrowania z wyprzedzeniem (write-ahead logging). System utrzymuje w pamięci trwałej strukturę danych nazywaną rejestrem (log). Każdy rekord rejestru opisuje jedną operację Pisania w transakcji i ma następujące pola: nazwa transakcji -nazwa transakcji wykonującej operację pisania, nazwa jednostki danych -nazwa zapisywanej jednostki danych, stara wartość -wartość jednostki danych przed zapisem, nowa wartość -wartość jednostki danych po zapisie. Rejestr zawiera też specjalne rekordy odnotowujące istotne zdarzenia występujące podczas przetwarzania transakcji, takie jak początek transakcji, jej zatwierdzenie lub zaniechanie.

Punkty kontrolne
Aby zmniejszyć dodatkowe koszty, wprowadza się punkty kontrolne (checkpoints).System podczas działania rejestruje z wyprzedzeniem operacje Pisania i co pewien czas organizuje punkty kontrolne, w których należy wykonać następujący ciąg czynności:
1. wszystkie rekordy aktualnie będące w PAO są zapisywane w pamięci trwałej.

2. wszystkie zmienione dane w pamięci ulotnej są umieszczane w pamięci trwałej.
3. w rejestrze transakcji (zawartym w pamięci trwałej) należy zapisać rekord <punkt kontrolny>.

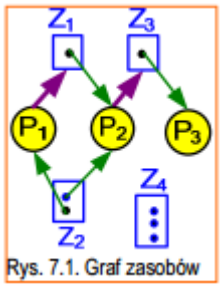
71. Omówić plan szeregowy w problematyce współbieżnych transakcji niepodzielnych.

Plan szeregowy: System zawiera dwa obiekty danych A i B, które są czytane i zapisywane przez transakcje T0 i T1.Transakcje wykonywane są niepodzielnie i transakcja T0 poprzedza transakcję T1.Plan szeregowy (serial schedule): każda transakcja wykonywana jest niepodzielnie. Jest poprawny, gdyż jest równoważny niepodzielnemu wykonaniu poszczególnych składowych transakcji w pewnym dowolnym porządku. Jeżeli instrukcje wchodzące w skład dwu transakcji będą się przeplatać, to wynikowy plan przestaje być szeregowy. ***Nie szeregowy*** plan może być w ostatecznym wyniku wykonania równoważny szeregowemu.

72. Podać niezbędne warunki prowadzące do zakleszczenia.

Zakleszczenie (deadlock): oczekujące procesy nie zmieniają swego stanu, gdyż zamawiane przez nie zasoby przydzielono innym procesom. Tylko jednoczesne zajście 4-ch warunków może prowadzić do zakleszczenia:
1. wzajemne wykluczanie: przynajmniej jeden zasób musi być niepodzielny (tylko jeden proces moze go używać w danym czasie).Jeśli inny proces zamawia dany zasób, to musi być opóźniany do czasu zwolnienia zasobu.
2. przetrzymywanie i oczekiwanie: istnieje proces, któremu przydzielono co najmniej jeden zasób i który oczekuje na przydział innego zasobu, przetrzymywanego przez inny proces.
3. brak wywłaszczeń: zasoby nie podlegają wywłaszczaniu - zasób moze zostać zwolniony tylko z inicjatywy przetrzymującego go procesu, po zakończeniu pracy tego procesu.
4. czekanie cykliczne: istnieje zbiór {P0, P1, ..., Pn} czekających procesów, takich ze P0 czeka na zasób przetrzymywany przez proces P1, P1 => P2, ..., Pn-1 => Pn oraz Pn => P0.

73. Graf przydziału zasobów – omówić, podać przykład, cel stosowanie.



Rys. 7.1. Graf zasobów

Graf przydziału zasobów systemu (system resource-allocation graph) jest grafem skierowanym, pozwalającym opisywać zakleszczenia. Można wykazać –w oparciu o definicję grafu przydziału zasobów- ze jeśli graf nie zawiera cykli, to w systemie nie ma zakleszczonych procesów.

74. Metody postępowania z zakleszczeniami.

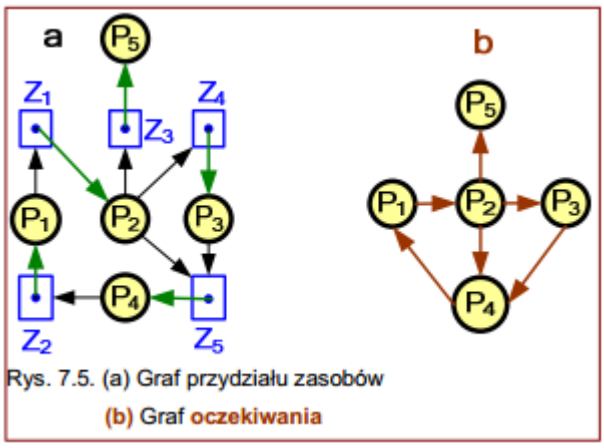
1. Zastosować protokół gwarantujący, ze system nigdy nie wejdzie w stan zakleszczenia.
Zapobieganie zakleszczeniom: zbiór metod zapewniających, ze co najmniej jeden z warunków koniecznych do wystąpienia zakleszczeń nie będzie spełniony. Metody nakładają ograniczenia na sposób zamawiania zasobów. Unikanie zakleszczeń: SO dysponuje dodatkowymi informacjami o zasobach, które proces będzie zamawiał. Każde zamówienie wymaga, aby system, podejmując decyzję czy można je zrealizować czy odłożyć, wziął pod uwagę aktualnie dostępne zasoby.
2. Zezwolić na zakleszczenia, po czym usunąć je. System dopuszczający wystąpienie zakleszczeń powinien

umożliwić:
-sprawdzenie jego stanu, aby określić, czy doszło do zakleszczenia,
-zlikwidować zakleszczenie jeśli wystąpiło.
3. Przyjąć założenie, ze zakleszczenia nigdy nie pojawią się w systemie.
System nie testuje zakleszczeń, wówczas będąc w stanie zakleszczenia, nie będzie o tym wiedział. Zakleszczenie pogorszy działanie systemu, gdyż coraz więcej procesów wykonujących zamówienia zasobów będzie ulegać zablokowaniu. W wielu systemach do zakleszczeń dochodzi rzadko. Przykładem może być proces czasu rzeczywistego wykonywany z najwyższym priorytetem (lub proces w systemie bez wywłaszczeń), który nie oddaje sterowania SO.

75. Zapobieganie, unikanie, likwidowanie zakleszczeń.

Warunek zakleszczenia to spełnienie każdego z czterech warunków. Gwarantując niespełnienie przynajmniej jednego z warunków można zapobiegać zakleszczeniom. Wzajemne wykluczanie. Warunek wzajemnego wykluczania musi być spełniony dla zasobów niepodzielnych. Przetrzymywanie i oczekiwanie Warunek ten nie wystąpi jeżeli gwarantujemy, ze proces zamawiający zasób, nie będzie miał żadnych innych zasobów. Brak wywłaszczeń Proces ma zasoby i zgłasza zapotrzebowanie na nowy zasób, którego nie można natychmiast przydzielić (proces musiałby czekać),Czekanie cykliczne Czekanie cykliczne nie wystąpi, po uporządkowaniu wszystkich typów zasobów i zamawianiu ich przez każdy proces we wzrastającym porządku ich numeracji. Ubocznym skutkiem zapobiegania zakleszczeniom może być słabe wykorzystanie urządzeń i ograniczona przepustowość systemu. Metoda unikania zakleszczeń wymaga dodatkowych informacji o sposobie zamawianie zasobów. Algorytm unikania zakleszczenia (dead lock avoidance) sprawdza dynamicznie stan przydziału zasobów, aby zagwarantować, ze nigdy nie dojdzie do spełnienia warunku czekania cyklicznego.
Likwidowanie zakleszczenia:
1. Usunięcie jednego lub kilku procesów w celu przerwania czekania cyklicznego. Zaniechanie wszystkich zakleszczonych procesów. Usuwanie procesów pojedynczo, aż do wyeliminowania cyklu zakleszczenia.
2.Odebranie pewnych zasobów procesom, których dotyczy zakleszczenie. Stopniowo odbiera się zasoby jednym procesom i przydziela innym, aż do przerwania cyklu zakleszczenia.

76. Metodyka wykrywania zakleszczeń, przykład grafowy.



Rys. 7.5. (a) Graf przydziału zasobów
(b) Graf oczekiwania

Stosowany tylko w przypadku gdy zasoby mają tylko jeden egzemplarz.. W grafie oczekiwania krawędź oznacza że proces czeka na zwolnienie zasobu wykorzystywanego przez inny proces. Zakleszczenie w cyklu wystąpi tylko gdy w grafie wystąpi cykl. Rząd operacji algorytmu wykrywania cyklu to n^2 gdzie n liczba wierzchołków grafu. Wywoływanie algorytmu przy każdym zamówieniu powoduje wydłużenie czasu obliczeń. Można zamiast tego wywoływać go w różnych odstępach czasu(gdy np. użycie CPU spadnie poniżej 40%), ale może to prowadzić do powstania wielu cykli i nie będzie można odnaleźć „sprawcy" powstania zakleszczenia.

79. Omówić pojęcie: semantyka spójności.

Semantyka spójności jest kryterium oceny systemu plików realizującego dzielenie plików. Określa warunki, przy których zmiany danych wykonywane przez jednego użytkownika są obserwowalne przez innych użytkowników.

80. Omówić mechanizmy przekazywania sterowania do procedury obsługi przerwań.

Dwa mechanizmy przekazywania sterowania do procedury obsługi przerwania: Wywołanie ogólnej procedury sprawdzającej informacje opisujące przerwanie; Wywołanie szczegółowej procedury obsługującej przerwanie. Utworzenie Tablicy wskaźników zwanej wektorem przerwań (interrupt vector) do procedur obsługujących przerwania, przy założeniu, ze liczba możliwych przerwań jest z góry znana. Tablica jest indeksowana jednoznacznym numerem urządzenia, które wysłało przerwanie. Zapewnia to właściwy adres do procedury obsługi przerwania, zgłoszonego przez urządzenie. Współczesne systemy operacyjne obsługują przerwania poprzez wektor przerwań.

81. Omówić mechanizmy bezpośredniej komunikacji między procesowej

Komunikacja międzyprocesowa (InterProcess-Communication –IPC) to oprogramowanie, umożliwiające procesom łączność i synchronizowanie działań. Komunikację najlepiej realizuje się za pomocą przekazywania komunikatów. System komunikatów umożliwia procesom wzajemną komunikację bez zmiennych dzielonych. Podstawowe operacje: Nadaj(komunikat) i Odbierz(komunikat).

82. Wyjaśnij zastosowanie mechanizmu blokowania stron w pamięci.

Umożliwienie blokowania stron w pamięci:. Każdej ramce przyporządkowuje się bit blokowania. Jeśli ramka jest zablokowana, to nie bierze udziału w zastępowaniu stron. Przed zapisaniem bloku na dysku blokuje się w pamięci zawierające go strony. Dalsze działanie systemu przebiega bez zmian. Stron zablokowanych nie można zastępować, zwalnia się je dopiero po zakończeniu operacji. Bit blokowania umożliwia ochronę nowo sprowadzonej strony przed zastąpieniem do czasu przynajmniej jednokrotnego jej użycia. Po wybraniu strony do zastąpienia jej bit blokowania otrzyma wartość 1 i zachowa ją tak długo, aż oczekującemu procesowi zostanie znów przydzielony procesor. Używanie bitu blokowania może być niebezpieczne, gdy po jego ustawieniu nie nastąpi zerowanie. + Zablokowana ramka nigdy nie będzie już udostępniona. System operacyjny komputera Macintosh stosuje blokowanie stron, gdyż jest to system dla jednego użytkownika i nadmiar zablokowanych stron zaszkodziłby tylko jednemu. System SunOS opuszcza „zalecenia" blokowania, które można odrzucić, jeśli pula wolnych stron staje się zbyt mała lub jeśli dany proces próbuje zablokować zbyt wiele stron w pamięci.

83. Wyjaśnij pojęcie domena ochrony, metody realizacji.

Proces działa w domenie ochrony, która określa dostępne dla niego zasoby. Domena definiuje zbiór obiektów i rodzaje operacji, które można wykonywać dla każdego obiektu. Prawo dostępu – możliwość wykonania operacji na obiekcie.
Metody realizacji domeny:
a) Użytkownik – jest domeną. Obiekty, do których można mieć dostęp, zależą od identyfikacji użytkownika. Przełączanie domen następuje wraz ze zmiana użytkownika.

b) Proces – jest domeną. Obiekty, do których jest dostęp zależą od identyfikacji procesu. Przełączanie domen to wysyłanie przez proces komunikatu do innego procesu.
c) Procedura – jest domeną. Obiekty, do których jest dostęp to zmienne lokalne w procedurze. Przełączanie domen następuje wraz z wywołaniem procedury.
Proces działający w domenie Użytkownika może wykonywać tylko rozkazy nieuprzywilejowane. Proces z domeny Monitora może wykonywać rozkazy. uprzywilejowane.

84. Co to jest macierz dostępów, zastosowanie, struktura.

Tabela 12.1. Macierz dostępów

Obiekt / Domena	F1	F2	F3	Printer
D_1	Czytaj		Czytaj	
D_2				Drukuj
D_3		Czytaj	Wykonaj	
D_4	Czytaj Pisz		Czytaj Pisz	

Wiersze macierzy dostępów reprezentują domeny, a kolumny – obiekty. Każdy element macierzy zawiera zbiór praw dostępu. Obiekty definiują jawnie nazwy kolumn i można pominąć nazwę Obiektu w prawie dostępu. Macierzy dostępów umożliwia realizacje decyzji politycznych dotyczących ochrony. Decyzje te dotyczą praw, które powinny znaleźć się w elemencie (i,k) tej macierzy. Należy określić domenę działania każdego procesu; jest to na ogol w gestii SO. Macierz dostępów pozwala implementować kontrole statyczna i dynamiczna związków miedzy procesami a domenami. Jeśli przełączymy proces z jednej domeny do drugiej, to wykonujemy operacje (Przełącz) na obiekcie (domenie). Zaliczenie domen do obiektów macierzy dostępów pozwala kontrolować przełączanie domen. Zmiana macierzy dostępów oznacza operacje na obiekcie, jakim jest macierz dostępów. Zmiany można nadzorować, gdyż macierz dostępów stała się obiektem macierzy dostępów. Każdy element macierzy dostępów można zmieniać z osobna, należy traktować każdy element tej macierzy jako obiekt podlegający ochronie. Przykładowa macierz dostępów zawiera cztery domeny i cztery obiekty: 3 pliki i drukarkę.

85. Przedstawić metody implementacji macierzy dostępów.

a) Tablica globalna – składa się ze zbioru trojek <domena, obiekt, zbiór_praw>. Tablica ta jest zazwyczaj duża i nie może znajdować się w PAO.
b) Wykazy dostępów do obiektów – Każda kolumna w macierzy dostępów może wykazem dostępu do danego obiektu. Obiektowi odpowiadają pary <domena, zbiór_praw>, które definiują wszystkie domeny ze zbiorami praw dostępu do danego obiektu.
c) Wykazy uprawnień do domen – jest spisem obiektów i operacji dozwolonych do wykonania na tych obiektach. Obiekt reprezentuje jego nazwa fizyczna lub adres zwany uprawnieniem.
d) Mechanizm zamka-klucza – każdy obiekt ma wykaz jednoznacznych wzorców binarnych, zwanych zamkami. Każda domena ma wykaz jednoznacznych wzorców binarnych, zwanych kluczami. Proces działający w domenie może mieć dostęp do obiektu tylko wtedy, gdy dana domena zawiera klucz pasujący do jednego z zamków tego obiektu. Klucze można przekazywać od domeny do domeny. Przywileje dostępów można uaktualniać poprzez zmiany niektórych kluczy przypisanych do obiektu.

86. Przedstawić problematykę ochrony na poziome języka programowania.

Ochrona nie może zajmować się wyłącznie projektant systemu operacyjnego. Ochrona powinna być narzędziem dostępnym dla projektanta aplikacji, aby zasoby podsystemu użytkowego mogły być strzeżone przed nadużyciami lub skutkami błędów. Określenie żądanej kontroli dostępu do dzielonego zasobu w systemie sprowadza się do napisania odpowiedniej deklaracji w odniesieniu do danego zasobu. Taka deklarację można dołączyć do języka programowania przez rozszerzenie jego możliwości operowania

typami. Zalety podejścia programowego:
- zapotrzebowanie na ochronę są deklarowane a nie programowane w formie ciągu wywołań SO
- wymagania dotyczące ochrony można sformułować niezależnie od środków dostarczanych przez konkretny SO
- projektant podsystemu nie musi dostarczać środków wymuszania ochrony
- Notacja deklaratywna jest naturalna ponieważ przywileje dostępów pozostają w ścisłym związku z lingwistyczna koncepcją danych
Bezpieczeństwo realizowane drogą programowa nie będzie tak duże jak gwarantuje jądro SO, ponieważ mechanizm programowy przyjmuje więcej założeń odnośnie działania systemu.

87. Czynności prowadzące do utworzenia procesu.

1. Utworzenie identyfikator procesu.
W głównej tablicy procesów zostaje założona nowa pozycja.

2. Alokacja przestrzeni adresowej dla procesu.
Pamięć potrzebna na programy i dane oraz stosu użytkownika (mogą być wyznaczone domyślnie lub przez użytkownika podczas tworzenia zadania). Dla procesu potomnego, proces rodzicielski przekazuje niezbędne wartości w parametrze polecenia utworzenia procesu. Muszą powstać odpowiednie powiązania do istniejących obszarów współdzielonych. Na koniec SO alokuje przestrzeń przeznaczoną dla bloku kontrolnego procesu.

3. Inicjalizacja bloku kontrolnego procesu.
Większość elementów części poświęconej informacjom o stanie procesora jest inicjowana z wartościami zero. Licznik rozkazów, ustawiany na pozycję początku programu, wskaźnik stosu systemowego określa granice stosu związanego z procesem. Elementy grupy informacji sterujących procesem inicjowane są standardowymi wartościami domyślnymi.

4. Ustawienie odpowiednich połączeń.
Jeśli SO wykorzystuje do szeregowania kolejki implementowane jako listy, wówczas wskaźnik do nowego procesu musi trafić do odpowiedniej kolejki.

5. Tworzenie pomocniczych struktur danych.
Dla każdego procesu SO może utrzymywać plik rozrachunkowy, przechowujący dane potrzebne do różnorodnych rozliczeń i tworzenia statystyk wydajności.

88. Omówić przyczyny przerwania wykonywania procesu.

1. Polecenie administracyjne - bezpośrednie zadanie wywołania funkcji systemu operacyjnego Gdy np.: w trakcie wykonywania procesu zostanie uruchomiony rozkaz wykonania operacji We/Wy (otwarcia pliku). Wywołanie podobnego rozkazu przekazuje sterowanie do procedury obsługi urządzenia We/Wy, będącej częścią kodu SO. Na ogół polecenia administracyjne powodują przeniesienie procesu do stanu Zablokowany.
2. Przerwanie - zewnętrzne zdarzenie w stosunku aktualnie wykonywanych czynności zegarowe -system operacyjny decyduje, kiedy aktualnie przetwarzany proces wyczerpie cały limit czasu przeznaczony na bieżący etap jego działania. Gdy to nastąpi, proces musi zostać przełączony w stan Gotowy, a SO wyznacza do pracy kolejny program; We/Wy -system operacyjny analizuje wszystkie zachodzące operacje We/Wy. Jeśli operacja osiągnie stan odpowiadający zdefiniowanemu zdarzeniu, wówczas SO przenosi wszystkie

oczekujące na to zdarzenie procesy ze stanu Zablokowany do stanu Gotowy. Następnie SO rozstrzyga, czy wznowi działanie aktualnie przetwarzanego procesu, czy tez wywłaszczy go, na rzecz, procesu o wyższym priorytecie; wskutek nietrafionych odwołań do pamięci -procesor natrafia co jakiś czas na odwołania do adresów pamięci wirtualnej odnoszące się do słów, których nie ma w PAO. SO musi wtedy przenieść odpowiedni blok danych z pamięci dyskowej do PAO. Po uruchomieniu operacji We/Wy, SO może przenieść bieżący proces w stan Zablokowany i wznowić przetwarzanie innego procesu. Kiedy odpowiedni blok danych znajdzie się już w PAO, przerwany proces zostanie przeniesiony do stanu Gotowy.
3. Pułapka **-** problem aktualnie wykonywanych czynności System operacyjny sprawdza, czy błąd lub warunek wyjątku mają charakter krytyczny. Jeśli tak, aktualnie wykonywany proces zostaje przeniesiony w stan Zakończony, a sterowanie przełączone do innego procesu. Jeżeli nie, to dalsze działanie systemu SO zależy od jego budowy i charakteru błędu. Mogą być uruchomione procedury odtworzeniowe, lub tez system tylko poinformuje użytkownika odpowiednim komunikatem. SO może uruchomić inny proces lub wznowić wykonywanie przerwanego programu.

Zadania z numerami pod kolorem to pytania najczęściej się powtarzające na egzaminach.