

\001(1).txt

1. W hierarchicznym systemie plików

- występuje wiele katalogów
- występuje wiele katalogów głównych
- użytkownikom przypisane są priorytety dostępu do plików
- Użytkownicy mają dostęp tylko do swoich plików
- Określa się ścieżki dostępu do plików

\001(2).txt

1. W hierarchicznym systemie plików

- występuje jeden katalog główny
- Określona jest dokładnie jedna ścieżka bezwzględna do pliku
- Występują ścieżki względne i bezwzględne dostępu do plików
- Moga występować wielokrotne dowiązania twarde do pliku

\002.txt

2. Bloki alokacji (klastry) w systemie plików

- Dobierane są tak, aby zminimalizować fragmentację zewnętrzną
- Dobierane są tak, aby zminimalizować fragmentację wewnętrzną
- Są stałej wielkości w całym systemie plików
- Są zawsze wielkości dwóch rekordów fizycznych (sektorów) na dysku
- Powinny być jak największe, aby było ich w systemie niewiele
- Numerowane są liczbami binarnymi stałej długości, co ogranicza ich liczbę

\003(1).txt

3. Sekwencyjny dostęp do pliku

- Polega na czytaniu lub zapisie pliku rekord po rekordzie w stałej kolejności
- Pozwala na występowanie plików nieciągłych, z niezaalokowanymi dziurami
- Jest możliwy tylko w plikach z rekordami stałej długości
- Nie pozwala dopisywać rekordów na koniec istniejącego pliku
- Nie powinien być używany - jest implementowany jedynie z przyczyn historycznych

\003(2).txt

3. Sekwencyjny dostęp do pliku

- Pozwala czytać rekordy w dowolnej kolejności dzięki operacji lseek
- Wyklucza powstawanie "dziur" w plikach, w których nie ma przydzielonych bloków
- Ułatwia grupowanie operacji dyskowych przez system operacyjny
- Powinien być używany jedynie przy operacjach na plikach specjalnych, odpowiadających urządzeniom znakowym
- Szereguje dostęp wielu procesów do pliku

\004.txt

4. Bezpośredni==swobodny dostep do pliku

- Wymaga zmiany kolejności rekordów w pliku w celu optymalizacji czasu dostępu
- Pozwala na występowanie plików nieciągłych, wewnątrz których mogą wystąpić niezaalokowane obszary
- Jest możliwy tylko w plikach z rekordami stałej długości
- Pozwala na występowanie plików nieciągłych, z niezaalokowanymi dziurami
- Pozwala czytać rekordy w dowolnej kolejności dzięki operacji lseek
- Pozwala na zapis fragmentów różnych plików do wspólnego bloku alokacji
- Szereguje dostęp wielu procesów do pliku
- Blokuję równoczesny dostęp wielu procesów do pliku

\005.txt

5. W Unixie występuje katalog, dostępny przez ścieżkę ../

- Jest to katalog roboczy procesu
- Jest to katalog nadrzędny
- Jest to katalog główny, jeden dla całego systemu
- Każdy proces może mieć swój taki katalog
- Proces musi mieć prawa zapisu w tym katalogu, aby mógł się wykonywać

\006.txt

6. Przydział ciągły miejsca na dysku

- Jest algorytmem optymalnym, nie dającym się zaimplementować w praktyce
- Charakteryzuje się fragmentacją zewnętrzną przestrzeni dysku
- Wymaga zapisu tablicy indeksowej na dysku

\007.txt

7. Przydział indeksowy miejsca na dysku

- Jest algorytmem optymalnym, nie dającym się zaimplementować w praktyce
- Charakteryzuje się fragmentacją zewnętrzną przestrzeni dysku
- Wymaga zapisu tablicy indeksowej na dysku

\008.txt

8. Przydział listowy miejsca na dysku

- Jest algorytmem optymalnym, nie dającym się zaimplementować w praktyce
- Charakteryzuje się brakiem fragmentacji wewnętrznej przestrzeni dysku
- Wymaga zapisu tablicy listowej na dysku
- Żadna z powyższych odpowiedzi nie jest prawdziwa

\009.txt

9. Planowanie krótkoterminowe

- Wymaga architektury wieloprocesorowej

- Bezpośrednio decyduje o wyborze procesów, które należy wczytać do pamięci
- Określa procesy, które należy zablokować
- Żadna z powyższych odpowiedzi nie jest prawdziwa

\010.txt

10. Planowanie średnioterminowe

- Wymaga architektury wieloprocesorowej
- Bezpośrednio decyduje o wyborze procesów, które należy wczytać do pamięci
- Określa procesy, które należy zablokować
- Polega na wyborze procesu, który ma zostać usunięty z pamięci op
- To wybór zadania, które w pierwszej kolejności powinno zostać wczytane do pamięci op (nie wiadomo, dalem jako prawda)
- Musi być zaimplementowane w każdym wieloprogramowym SO

\011.txt

11. Planowanie długoterminowe

- Wymaga architektury wieloprocesorowej
- Bezpośrednio decyduje o wyborze procesów, które należy wczytać do pamięci
- Określa procesy, które należy utworzyć

\012.txt

12. "Spooling"

- To technika przydzielania dodatkowych buforów procesom uprzywilejowanym
- To technika kopiowania w tle danych z nosnika papierowego na szybszy magnetyczny
- To technika programowania, zmniejszająca pamięć wymagana do wykonania programu
- Eliminuje fragmentację zewnętrzną dysków przez przesuwanie plików w trakcie normalnej pracy komputera (w tle)
- Eliminuje fragmentację zewnętrzną dysków przez przesuwanie plików, przy czym uruchamiany jest przy wyłączonych wszystkich innych procesach
- To metoda buforowania dysku bazująca na tworzeniu dużych puli buforów systemowych
- To metoda buforowania dysku w szybkich pamięciach wbudowanych do sterownika
- Polega na zapisie danych wyjściowych z programu na szybkie nośniki magnetyczne i drukowaniu ich w tle po zakończeniu programu

\013.txt

13. W Unixie występuje katalog, dostępny przez ścieżkę ./

- Jest to katalog roboczy procesu
- Jest to katalog nadrzędny
- Jest to katalog główny, jeden dla całego systemu
- Każdy proces może mieć swój taki katalog
- Proces musi mieć prawa zapisu w tym katalogu, aby mógł się wykonywać

\014(1).txt

14. Katalog główny systemu plików w Unixie

- Jest plikiem zawierającym rekordy z nazwami i numerami inode'ów należących do niego plików
- Jest plikiem zawierającym rekordy z nazwami, innymi atrybutami i tablicami indeksowymi należących do niego plików
- Nie ma nazwy, definiowany jest przez zapisanie numeru jego inode'u w superbloku
- Nie ma nazwy, zajmuje zawsze to samo miejsce na partycji dysku (w systemie plików)
- Ma ograniczoną wielkość, ponieważ umieszczony jest w całości w superbloku

\014(2).txt

14. Katalog główny systemu plików w Unixie

- Nie zawiera pozycji .. (normalnie wskazującej na katalog nadrzędny)
- Nie zawiera pozycji . i .. (normalnie wskazujących na katalog bieżący i nadrzędny)
- Zawiera stałą liczbę pozycji, której nie można zmienić bez rekompilacji jądra systemu
- Musi mieć taką samą zawartość, jak katalog do którego jest montowany
- Żadna z powyższych odpowiedzi nie jest prawdziwa

\015.txt

15. Katalog bieżący w Unixie

- Jest plikiem zawierającym rekordy z nazwami i numerami inode'ów należących do niego plików
- Ma ograniczoną wielkość, ponieważ umieszczony jest w całości w superbloku
- Nie zawiera pozycji . i ..
- Użytkownik nie może zmienić go, jeżeli nie ma uprawnień Root

\016.txt

16. Powłoka systemu operacyjnego

- Określa dostępny zestaw i składnię komend sterujących
- Określa dostępny zestaw i składnię komend zewnętrznych
- Określa dostępny zestaw i składnię komend wewnętrznych
- Określa zmienne środowiskowe dla uruchamianych komend
- Określa język programowania skryptów systemowych
- Pozwala na edytowanie linii komendy
- Pozwala na przeglądanie historii wykonywanych komend

\017.txt

17. Listy kontroli dostępu ACL w Windows NT

- Służą do określania warunków logowania użytkowników do systemu
- Służą do określania warunków dostępu do plików dla różnych użytkowników
- Pozwalają definiować grupy użytkowników
- Pozwalają definiować klucze dostępu do zasobów
- Umożliwiają zdefiniowanie obiektów, do których dostęp musi być odnotowany

\018.txt

18. Plik MFT (Master File Table) w systemie plików NTFS

- Służy do określania uprawnień dostępu do plików
- Służy do określania warunków logowania użytkowników do systemu
- Zawiera listy ACL wszystkich plików
- Zawiera kopie boot sektora
- Zawiera opis wszystkich plików w systemie
- Zawiera tablice partycji dysku

\019.txt

19. Tablica FAT w systemie MS-DOS

- Zawiera osobną pozycję dla każdego rekordu fizycznego (sektora) w partycji dysku
- Zawiera informacje o defektach nośnika w pamięci dyskowej
- Pozwala przydzielać rekordy rezerwowe w miejsce uszkodzonych
- Zastępuje mapę zajętości dysku

\020.txt

20. Semafor binarny

- Służą do wyznaczania czasu w systemach czasu rzeczywistego
- Służą do synchronizacji procesów współbieżnych
- Służą do blokowania dostępu do dysku w trybie binarnym (nie znakowym)
- Są to struktury danych do modelowania ruchu ulicznego
- Są to dane jednobitowe, które można testować, ustawiać i zerować
- Są to dane całkowitoliczbowe z zakresu 0 - 2n (n>1) z niepodzielnymi operacjami P i V
- Można używać do rozwiązywania dowolnych problemów synchronizacji procesów

\021(1).txt

21. Semafor Dijkstry

- Operacja czekaj na tych semaforach zawsze usypia procesy
- Operacja sygnalizuj na tych semaforach może usypiać procesy
- Są mechanizmem ograniczającym dostęp do plików
- Służą do synchronizacji procesów
- Służą do synchronizacji procesów współbieżnych
- Są to struktury danych do modelowania ruchu ulicznego

\021(2).txt

21. Semafor Dijkstry

- Są mechanizmem synchronizującym dostęp do plików
- Są to dane całkowitoliczbowe, na których zdefiniowane są operacje P i V
- Są to flagi dwustanowe, na których zdefiniowane są nieprzerywalne operacje P i V
- Można używać do rozwiązywania dowolnych problemów synchronizacji procesów
- W ich implementacji stosuje się petle gorącego czekania

\022(1).txt

22. Kontekst procesu

- Okresla wszystkie zasoby, z których może proces korzystać
- Okresla wszystkie procesy, z którymi może się komunikować
- Okresla zawartość rejestrów, z których proces może korzystać
- Obejmuje informacje konieczne do wznowienia procesu zablokowanego
- Obejmuje informacje konieczne do ustalenia procesu nadrzędnego
- Jest to obraz pamięci procesu w pewnej chwili
- Jest to między innymi obraz pamięci procesu w chwili wstrzymania

\022(2).txt

22. Kontekst procesu

- Obejmuje informacje konieczne do wznowienia procesu wstrzymanego
- Jest to obraz plików otwartych przez proces w pewnej chwili
- Jest to obraz pamięci, rejestrów i zmiennych systemowych procesu w pewnej chwili
- Jest składowany gdy proces się kończy (wykonuje exit)
- Jest odtwarzany przy każdym przełączeniu procesów
- Opisuje zależność pomiędzy współbieżnymi procesami

\023.txt

23. Procesy charakteryzują się

- Sekwencyjnym wykonywaniem operacji
- Kontekstem, który określa jakie procesy są wykonywane współbieżnie
- Złożonością obliczeniową, wyrażoną np. przez liczbę pętli, skoków i rozgałęzień (Nie wiadomo do końca)
- Możliwością przerywania i wznowiania wykonania (według składowanego kontekstu)

\024.txt

24. Proces może przejść do stanu gotowości

- Po wyczerpaniu kwantu czasu aktywności (w systemie z algorytmem karuzelowym)
- W wyniku wykonania funkcji exit (w Unixie)
- Bezpośrednio ze stanu zablokowania

\025.txt

25. Proces może przejść do stanu zablokowania (uspiania)

- Po wyczerpaniu kwantu czasu aktywności (w systemie z algorytmem karuzelowym)
- W wyniku wykonania funkcji exit (w Unixie)
- Bezpośrednio ze stanu gotowości

\026.txt

26. Ochrona pamięci w systemach operacyjnych

- Ogranicza dostęp procesów do pamięci systemu operacyjnego
- Ogranicza dostęp procesów do pamięci im nie przydzielonej
- Zapewnia, że inne procesy nie mają dostępu do pamięci procesu
- Zapewnia, że procesy użytkowe nie mogą zmodyfikować obrazu pamięci systemu oper.

- Bazuje na przekształcaniu adresów logicznych na fizyczne, kontrolowanym przez SO
- Bazuje na szyfrowaniu zawartości pamięci
- Określa prawa dostępu procesów do plików różnych właścicieli
- Określa prawa dostępu procesów do folderów poczty elektronicznej

\027.txt

27. Stronicowanie na zadanie

- To podstawowa technika zwiększania pamięci widzianej przez programy
- To podstawowa technika przyspieszenia dostępu do pamięci
- Polega na składowaniu na dysk całego obrazu pamięci zablokowanego procesu
- Polega na buforowaniu stron pamięci w pamięci podręcznej cache
- Wymaga dodatkowych pól w tablicy stron (np. flagi ważności)
- Wymaga utrzymywania w pamięci tablicy stron

\028.txt

28. Stronicowanie pamięci

- Polega na przydziale do programu ramek pamięci o stałej wielkości
- Polega na przydziale do programu segmentów pamięci dowolnej wielkości
- Wymaga tworzenia obrazu pamięci w obszarze wymiany na dysku
- Pozwala implementować liniowy model adresów fizycznych w programie
- Pozwala implementować liniowy model adresów logicznych w programie

\029(1).txt

29. Pamięć stronicowania charakteryzuje się tym, że

- Adresy logiczne są tworzone jako suma adresu bazowego i przesunięcia
- Adresy fizyczne są sklejaniem pary liczb określających numer ramki stałej długości oraz przesunięcia (offset) w obrębie ramki
- Adres logiczny jest złożeniem numeru strony i przesunięcia względem jej początku
- Jeżeli strona nie ma przydzielonej ramki fizycznej, to występuje błąd stronyE; brak przydziału jest oznaczony w tablicy stron flagą poprawności (valid)
- Jeżeli strona nie ma przydzielonej ramki fizycznej, to występuje błąd stronyE; brak przydziału jest oznaczony w tablicy stron bitem ochrony (access control)
- Jeżeli strona nie ma przydzielonej ramki fizycznej, to występuje błąd stronyE; brak przydziału jest oznaczony w tablicy stron bitem modyfikacji (modified flag)
- Jest technika pozwalająca przydzielać programom więcej pamięci, niż jest fizycznie zainstalowane w systemie
- Jest technika pozwalająca przydzielać programom pamięć blokami stałej wielkości (stronami)

\029(2).txt

29. Pamięć stronicowania charakteryzuje się tym, że

- Adres logiczny jest przekształcany na adres fizyczny przez odczytanie numeru ramki z tablicy stron, z pozycji odpowiadającej numerowi strony
- Jest technika pozwalająca przydzielać programom pamięć blokami zmiennej wielkości
- Eliminuje fragmentację zewnętrzną pamięci
- Eliminuje fragmentację wewnętrzną pamięci

- Musi być realizowana z wykorzystaniem pamięci asocjacyjnej na tablicy stron
- Powoduje, że programy mają nieliniową przestrzeń adresową (fizyczna T.AK, logiczna N.IE).
- Wymaga implementacji tablicy segmentów (sprzętowo lub programowo)

\030.txt

30. Blokada systemu

- występuje, gdy dwa procesy próbują równocześnie pisać do tego samego pliku
- występuje, gdy dwa procesy próbują równocześnie pisać do tego gniazda sieciowego
- występuje tylko w systemach operacyjnych napisanych z błędami
- występuje tylko w systemach operacyjnych wieloprogramowych
- Mozna ją usunąć przez wymuszenie zakończenia wszystkich procesów uspionych
- Można ją usunąć przez czasowe zablokowanie dostępu użytkowników do systemu

\031(1).txt

31. Warunki konieczne do wystąpienia blokady to

- Występowanie zasobów, które nie mogą być równocześnie wykorzystywane przez wiele procesów
- Brak priorytetów w dostępie do zasobów
- Przełączanie procesów według algorytmu planowania z wywłaszczaniem
- Przełączanie procesów według algorytmu bez wywłaszczania
- Przetrzymanie i oczekiwanie, tzn. pozwolenie procesom nie będącym aktywnymi ani gotowymi na zachowanie przydziału zasobu
- Brak mechanizmu, który umożliwiłby procesom przyjęcie zasobów przydzielonych innym

\031(2).txt

31. Warunki konieczne do wystąpienia blokady to

- Występowanie zasobów, które nie mogą być współdzielone
- Brak mechanizmu umożliwiającego wywłaszczanie zasobów od procesów
- Określają, jak pisać programy wielowatkowe aby uniknąć blokad
- Określają, jak pisać programy wieloprotokółowe aby uniknąć blokad
- Określają, kiedy proces może przejść do stanu zablokowania
- Możliwość zawieszania procesów z przydzielonymi pewnymi zasobami

\032.txt

32. Funkcja systemowa wait w Unixie

- Zawiesza wykonywanie procesu w oczekiwaniu na zakończenie procesu potomnego
- Zawiesza wykonywanie procesu przez zadany argumentem czas
- Może być wykonywana tylko przez procesy demony
- Może być wykonywana tylko przez proces z uprawnieniami root
- Umożliwia czekanie na zakończenie operacji sieciowych
- Umożliwia czekanie na zakończenie operacji we/wy

\033.txt

33. Technika opóźnionego zapisu na dysk w systemie Unix

- Jest metu sieciowym
- Jest metoda synchronizacji zapisów na dysku
- Polega na wprowadzeniu petli opóźniajacej miedzy kolejnymi rozkazami zapisu
- Polega na zapisie bufora na dysk, w chwili jego zwalniania
- Moze byc wybiórczo stosowana przez programowe włączanie/wyłączanie funkcja fctl
- Moze byc wyłączana komenda fctl

\034.txt

34. Buforowanie operacji dyskowych w systemie Unix

- Jest ograniczone do urzadzen z niewymiennym nosnikiem (np. twardych dysków); dostep do plików na dyskietkach nie moze byc buforowany
- Redukuje czas dostępu do plików oraz liczbe fizycznych operacji we/wy.
- Utrudnia równoczesny dostep do rekordów pliku przez wiele procesów (recordlocking)
- Wymaga od programisty uzywania operacji fflush aby miec pewnosc, ze zmiany w pliku zostana natychmiast zapisane na dysk

\035.txt

35. Sygnały w systemie Unix

- Wykorzystywane sa do synchronizacji procesów
- Wykorzystywane sa m.in. do zabijania procesów
- Moga byc adresowane do wszystkich procesów uzytkownika o znanym UID
- Moga byc adresowane do wszystkich procesów uspionych w systemie
- Mozna definiowac własne funkcje obsługi niektórych z nich
- Proces mozna zaprogramowac tak, aby ignorował wszystkie sygnały przychodzace
- Wysyłane moga byc przez kazdy proces przy pomocy funkcji signal
- Wysyłane moga byc przez kazdy proces przy pomocy funkcji kill

\036(1).txt

36. Fragmentacja wnetrzna dysku jest wynikiem

- Niedopasowania rozmiaru plików do wielkoscí obszarów wolnych (powstajacych przy kasowaniu plików)
- Niedopasowania rozmiaru plików do wielokrotnosci wielkoscí bloków alokacji
- Przydziału miejsca na dysku porcjami zmiennej wielkoscí
- Moze byc wyeliminowana przez program defragmentujacy
- Pozostawiania niewykorzystanych fragmentów w ostatnich blokach plików

\036(2).txt

36. Fragmentacja wnetrzna dysku jest wynikiem

- Przydziału miejsca na dysku porcjami stałej wielkoscí
- Braku konserwacji systemu plików - moze byc wyeliminowana przez program defragmentujacy
- Braku konserwacji systemu plików (brak defragmentacji)
- Błędów programowych np. nie zamknięcia pliku przed zakonczeniem programu

\037.txt

37. Fragmentacja wnetrzna pamieci jest wynikiem

- Powoduje lepsze wykorzystanie miejsca w pamięci przez procesy
- Przydziału pamięci porcjami stałej wielkości (stronami)
- występuje jedynie w systemach z pamięcią wirtualną
- Niedopasowania rozmiaru segmentów pamięci do wielokrotności wielkości ramki
- Może być wyeliminowana przez relokację programów
- Błędów programowych np. nie zwalniania pamięci przed zakończeniem programu

\038.txt

38. Fragmentacja zewnętrzna pamięci jest wynikiem

- Niedopasowania rozmiaru segmentów pamięci do wielokrotności wielkości ramki
- Może być wyeliminowana przez relokację programów
- Błędów programowych np. nie zwalniania pamięci przed zakończeniem programu

\039(1).txt

39. Algorytm przełączania procesów w systemie Unix

- Jest algorytmem ze stałym priorytetem zewnętrznym, zadawanym komendą/funkcją nice
- Jest algorytmem bez wyłączeń
- Jest algorytmem z wyłączeniem procesów systemowych i użytkowych
- Jest algorytmem, w którym wyłączane są tylko procesy użytkownika
- Jest algorytmem priorytetowym, w którym priorytet jest wewnętrznie zmieniany w trakcie wykonywania procesu

\039(2).txt

39. Algorytm przełączania procesów w systemie Unix

- Jest algorytmem z wyłączeniem
- Opiera się na priorytecie statycznym, wyliczanym w oparciu o wykorzystanie procesora przez proces
- Opiera się na priorytecie dynamicznym, obniżanym dla procesów intensywnych obliczeniowo, a podwyższanym dla procesów interakcyjnych
- Pozwala modyfikować priorytet procesu przy pomocy funkcji nice

\040.txt

40. Algorytm karuzelowy przełączania procesów

- Jest algorytmem bez wyłączenia
- Jest algorytmem z wyłączeniem
- Przełącza procesy według dynamicznie zmienianego priorytetu
- Jest najczęściej wykorzystywanym algorytmem w różnych wersjach systemu operacyjnego Unix
- Polega na uruchamianiu procesów na stały kwant czasu i wymuszaniu przełączenia kontekstu, jeżeli proces nie zwolni procesora w tym czasie
- Jest algorytmem optymalnym, w praktyce niemożliwym do zrealizowania

\041.txt

41. Algorytmy przełączania procesów z wyłączeniem

- Moga wykorzystywać priorytety stałe lub zmieniające się dynamicznie
- Zwiększają współczynnik wykorzystania procesora w systemach wielodostępnych
- Polega na uruchamianiu procesów na stałym kwocie czasu i wymuszaniu przełączenia kontekstu, jeżeli proces nie zwolni procesora w tym czasie

\042.txt

42. Algorytm FIFO zastępowania stron pamięci

- Gwarantuje najmniejszą możliwą liczbę błędów stron
- Gwarantuje najkrótszy czas dostępu do pamięci
- Zapewnia, że liczba błędów stron (dla tych samych procesów) nie zwiększy się, gdy doinstalujemy (udostępnimy) dodatkowe ramki pamięci fizycznej
- Polega na usuwaniu stron, które najdłużej przebywały w pamięci fizycznej
- Polega na usuwaniu stron, które najdawniej były używane

\043.txt

43. Algorytm LRU (najdawniej używana) zastępowania stron pamięci

- Gwarantuje najmniejszą możliwą liczbę błędów stron
- Zapewnia, że liczba błędów stron (dla tych samych procesów) nie zwiększy się, gdy doinstalujemy (udostępnimy) dodatkowe ramki pamięci fizycznej
- Polega na usuwaniu stron, które najdłużej przebywały w pamięci fizycznej

\044(1).txt

44. Systemy wielodostępne

- Wymagają architektury wieloprocessorowej
- Wymagają architektury jednoprocessorowej
- Pozwalają na wykonywanie wielu zadań współbieżnie
- Pozwalają na współbieżne przetwarzanie wielu procesów (jednego lub wielu użytkowników)
- Zapewniają dostęp wszystkim użytkownikom do identycznych danych

\044(2).txt

44. Systemy wielodostępne

- Pozwalają na przesyłanie sobie poczty w obrębie systemu
- Wymuszają pracę wsadową na komputerze
- Wymagają wielu terminali podłączonych do jednego komputera
- Dzieli zadania na interakcyjne i wykonywane w tle

\045.txt

45. Dynamiczne ładowanie programu

- Polega na wykorzystaniu bibliotek dołączanych w trakcie wykonywania programu
- Polega na ładowaniu nakładek programu, gdy są potrzebne
- Mozna zrealizować w systemie ze stronicowaniem na zadanie

\046.txt

46. Dynamiczne łączenie programu

- Polega na wykorzystaniu bibliotek dołączanych w trakcie wykonywania programu
- Polega na ładowaniu nakładek programu, gdy są potrzebne
- Mozna zrealizować w systemie bez stronicowania na zadanie

\047.txt

47. Kryteriami rozwiązania problemu sekcji krytycznej są

- Wzajemne wyłączenie - procesy muszą mieć możliwość wzajemnego blokowania
- Postęp - dostęp do sekcji nie może być zablokowany, jeżeli nie przebywa w niej inny
- Ograniczone oczekiwanie - każdy proces musi kiedyś dostać się do sekcji

\048.txt

48. Dołączenia w systemie plików (links)

- Oznaczone są symbolem / w ścieżkach dostępu
- Pozwalają tworzyć systemy plików o strukturze niehierarchicznej
- Są wynikiem błędów w programie i polegają na połączeniu się list alokacji odpowiadających różnym plikom w tablicy FAT
- Są wynikiem błędów w programie i powstają przy wielokrotnym otwieraniu tych samych plików do zapisu
- Wiąza nazwy plików z INODEjami w systemie Unix
- Tworzone są przy otwieraniu plików, usuwane przy zamykaniu
- Mogą być symboliczne, gdy wskazują ścieżkę dostępu do pliku właściwego, lub twarde, jeżeli powstają przez zdefiniowanie drugiej nazwy dla pliku

\049.txt

49. Przetwarzanie potokowe w Unixie

- Realizowane jest przez przekierowanie standardowego wyjścia komendy pierwszej do standardowego wejścia komendy drugiej
- W powłoce sh można je wywołać wypisując dwie komendy w nawiasach okrągłych, rozdzielone przecinkami
- Dostępne jest tylko w systemach wieloprocesorowych

\050.txt

50. Proces uspiony (zablokowany)

- Oczekuje na zwolnienie procesora przez proces aktywny
- Realizuje operacje semaforowe
- Zawsze usuwany jest z pamięci operacyjnej (kopiowany do obszaru wymiany)

\051.txt

51. Stronicowanie na zadanie

- Polega na wymianie stron w pamięci fizycznej, wczytywaniu stron, do których występują aktualnie odwołania
- Umożliwia zapisywanie do obszaru wymiany na dysku fragmentów kontekstów procesów, gdy brakuje pamięci fizycznej

\052.txt

52. Mechanizm spotkaniowy

- To technika synchronizacji dostępu do pamięci współdzielonej przez procesy
- Wykorzystywany jest do przesyłania danych pomiędzy procesami
- Wykorzystywany jest do negocjacji dostępu do plików

\053.txt

53. Warunek postępu w synchronizacji między procesami

- To zapewnienie, że dwa procesy nie znajdują się w tym samym czasie w sekcji krytycznej
- To wymaganie, aby synchronizacja nie powodowała nieograniczonego w czasie zawieszenia procesu
- Zapewnia, że proces nie jest wyłaszczany, gdy znajduje się w sekcji krytycznej

\054.txt

54. Planowanie zadań ma na celu

- Unikanie blokad
- Synchronizowanie dostępu do zasobów
- Wybór procesu do aktywacji

\055.txt

55. Funkcja systemowa fork w Unixie (Linuxie)

- Tworzy nowy proces, realizujący program podany w argumencie do tej funkcji
- Tworzy nowy wątek, realizujący funkcję podaną w argumencie
- Tworzy nowy proces, realizujący ten sam program co proces-rodzic

\056.txt

56. Sygnały w systemie Unix

- Generowane są wyłącznie przez sterowniki urządzeń zewnętrznych
- Wysyłane są do pojedynczych procesów, grup procesów, procesów określonego użytkownika lub do wszystkich procesów
- Mogą być ignorowane przez procesy (z wyjątkiem sygnału 9)

\057.txt

57. Pozycje katalogowe

- Zawierają nazwy plików
- Zawierają informacje o położeniu pliku na dysku

- Zawierają sumę kontrolną pliku

\058.txt

58. Ścieżka do pliku

- Może być względna, bezwzględna lub symboliczna
- Ścieżka względna określa położenie pliku względem katalogu domowego użytkownika
- Ścieżka bezwzględna określa położenie pliku względem katalogu głównego

\059.txt

59. Sekcja krytyczna

- To technika synchronizacji dostępu do pamięci współdzielonej przez procesy
- Wykorzystywana jest do ochrony danych przed równoczesną ich modyfikacją przez procesy współbieżne
- Wykorzystywana jest do negocjacji dostępu do semaforów

\060.txt

60. Współbieżność w systemach operacyjnych

- Realizowana jest przez przełączanie procesów, tzn. wstrzymywanie procesów co pewien czas i przełączanie innych
- W powłocie shell można ją wywołać wpisując dwie komendy w nawiasach okrągłych, rozdzielone przecinkami
- Dostępna jest tylko w systemach wieloprotocowych

\061.txt

61. Proces gotowy

- Oczekuje na zwolnienie procesora przez proces aktywny
- Realizuje operacje drugoplanowe
- Czasami usuwany jest z pamięci operacyjnej (kopiowany do obszaru wymiany)

\062.txt

62. Funkcja systemowa exec w Unixie (Linuxie)

- Tworzy nowy proces, realizujący program podany w argumencie do tej funkcji
- Tworzy nowy wątek, realizujący funkcję podaną w argumencie
- Modyfikuje kontekst wykonywanego procesu, korzystając z pliku wykonywalnego podanego w argumencie

\063.txt

63. Sygnały w systemie Unix

- Generowane są przez wykonanie operacji kill
- Mogą powodować zakończenie procesu
- Mogą być wysłane do wielu procesów równocześnie

\064.txt

64. Dowiązania symboliczne

- Są to pliki specjalne, które wskazują na inne obiekty w systemie plików
- Zawierają informacje o położeniu pliku na dysku
- W Unixie muszą mieć rozszerzenie .lnk

\065.txt

65. Wzorce plików

- Są mechanizmem definiowania dopuszczalnej zawartości plików
- Pozwalają definiować ścieżki wyszukiwania komend w systemach operacyjnych
- Wykorzystują znaki ? oraz * do określania dowolnych znaków w nazwie pliku

\066.txt

66. Funkcja systemowa execve w Unixie(Linuxie)

- Tworzy nowy proces, realizujący program podany w argumencie do tej funkcji
- Tworzy nowy wątek, realizujący funkcję podaną w argumencie
- Tworzy nowy proces, realizujący ten sam program co proces-rodzic
- Żadna z powyższych odpowiedzi nie jest prawdziwa

\067.txt

67. Pozycje katalogowe w systemie Unix

- Zawierają nazwy plików
- Zawierają datę modyfikacji pliku
- Zawierają rozszerzenie 3 - znakowe

\068.txt

68. Stan procesu

- Jest pamiętany w tablicy procesów
- Jest wyznaczany przy pierwszym uruchomieniu procesu i potem się nie zmienia
- Jeżeli jest aktywny, to proces jest wykonywany przez procesor

\069.txt

69. Sygnały w systemie Unix

- Generowane są wyłącznie przez sterowniki urządzeń zewnętrznych
- Wysyłane są do pojedynczych procesów lub do grup procesów
- Większość może być ignorowana przez procesy.

\070.txt

70. Kanał komunikacyjny z ograniczonym buforowaniem

- To technika synchronizacji dostępu do karty sieciowej przez procesy
- Wykorzystywany jest do przesyłania danych pomiędzy procesami
- Może spowodować uśpienie procesu piszącego do kanału

\071.txt

71. Warunek wykluczania w synchronizacji między procesami

- To zapewnienie, że dwa procesy nie znajdą się w tym samym czasie w sekcji krytycznej
- To wymaganie, aby synchronizacja nie powodowała nieograniczonego w czasie zawieszenia procesu
- Zapewnia, że proces nie jest wyłaszczany, gdy znajduje się w sekcji krytycznej

\072.txt

72. Funkcja systemowa signal w Unixie (Linuxie)

- Tworzy nowy proces
- Wysła komunikat do standardowego strumienia błędów
- Wysła sygnał do procesu lub grupy procesów
- Żadna z powyższych odpowiedzi nie jest prawdziwa

\01.txt

Algorytm karuzelowy:

- Zapewnia wszystkim procesom ten sam priorytet
- Nie pozwala wywłaszczać procesów systemowych
- Jest algorytmem z priorytetami dynamicznymi

\02.txt

Algorytm LRU zastępowania stron pamięci wirtualnej:

- Polega na usuwaniu najdłużej załadowanych stron
- Zapewnia dobre dostosowanie do zestawu roboczego stron
- Zapewnia zawsze najmniejszą liczbę błędów stron (w porównaniu z innymi algorytmami)

\03.txt

Biblioteki łączone dynamicznie (DLL):

- Ładowane są do pamięci tylko funkcje, do których wystąpi odwołanie
- Udostępniają wyłącznie funkcje wielowejsciowe (reentrant)
- Kopiowane są w całości do plików wykonywalnych, zawierających programy z nich korzystające

\04.txt

Buforowanie operacji dyskowych

- Eliminuje fragmentację wewnętrzną w systemie plików
- Umożliwia zapis danych do pliku porcjami różnej wielkości
- Umożliwia zrównoleglenie procesów

\05.txt

Dowiązania symboliczne w Linuxie:

- Zawierają informacje o położeniu pliku na dysku
- Muszą mieć rozszerzenie .lnk
- Są to pliki specjalne, które wskazują przy pomocy ścieżek na inne obiekty

\06.txt

Dynamiczne łączenie programu:

- Zmniejsza wielkość pliku wykonywalnego (exe)
- Wykorzystuje biblioteki statyczne
- Polega na tworzeniu połączeń dynamicznych pomiędzy procesami

\07.txt

Dynamiczne ładowanie programu:

- Polega na ładowaniu nakładek programu, gdy są potrzebne
- Polega na wykorzystaniu bibliotek dołączanych w trakcie wykonywania programu
- Zmniejsza czas od wydania komendy do uruchomienia programu

\08.txt

Funkcja systemowa signal:

- Tworzy nowy proces
- Wysła sygnał do procesu lub grupy procesów
- Wysła komunikat do standardowego strumienia błędów

\09.txt

Funkcja systemowa fork w Unicie (Linuxie):

- Tworzy nowy proces, realizujący ten sam program co proces-rodzic
- Tworzy nowy wątek, realizujący funkcję podaną w argumencie
- Tworzy nowy proces, realizujący program podany w argumencie do tej funkcji

\10.txt

Funkcja systemowa execve w Unixie (Linuxie)

- Ładuje nowy program w miejsce obrazu pamięci procesu, który wywołał tę funkcję
- Tworzy nowy proces, realizujący program podany w argumencie do tej funkcji
- Tworzy nowy wątek, realizujący funkcję podaną w argumencie

\11.txt

Kanał komunikacyjny z zerowym buforowaniem:

- Umożliwia buforowanie operacji dyskowych
- To kanał, w którym aktualnie nie ma żadnych komunikatów
- Może powodować uśpienie procesu czytającego z kanału

\12.txt

Kontekst procesu:

- Jest to obraz pamięci, rejestrów i zmiennych systemowych procesu w pewnej chwili
- Obejmuje informacje konieczne do zablokowania procesu
- Opisuje stan wszystkich zasobów, z których może proces korzystać

\13.txt

Kanał komunikacyjny z nieograniczonym buforowaniem:

- Może powodować uśpienie procesu piszącego do kanału
- Może powodować uśpienie procesu czytającego z kanału
- Umożliwia buforowanie operacji dyskowych

\14.txt

Ogółle semaforey Dijkstry:

- Przyjmują tylko wartości 0 lub 1
- Mogą blokować proces przy wykonaniu operacji V
- Mogą blokować proces przy wykonywaniu operacji P

\15.txt

Ochrona pamięci w systemach operacyjnych:

- Określa prawa dostępu procesów do pamięci dyskowej
- Zapewnia, że procesy nie zostaną zmodyfikowane w trakcie kopiowania do obszaru wymiany (swapping)
- Uniemżliwia dostęp procesu do niezaalokowanej pamięci

\16.txt

Przetwarzanie pośrednie:

- To technika przydzielania dodatkowych buforów procesom uprzywilejowanym
- Eliminuje fragmentację zewnętrzną dysku przez przesuwanie plików w tle
- Polega na kopiowaniu danych między dyskiem a wolnymi urządzeniami zewnętrznymi w celu poprawy wykorzystania procesora

\17.txt

Przetwarzanie potokowe:

- Dostępne jest tylko w systemach wieloprocessorowych
- W powłoce sh można je wywołać wypisując dwie komendy w nawiasach okrągłych rozdzielone przecinkami
- Realizowane jest przez przekierowanie standardowego wyjścia komendy pierwszej do standardowego wejścia komendy drugiej

\18.txt

Przydział indeksowy miejsca na dysku:

- Charakteryzuje się fragmentacją zewnętrzną przestrzeni dysku
- Wymaga zapisu tablicy indeksowej na dysku
- Opisuje położenie pliku jako ciąg segmentów różnej wielkości

\19.txt

Proces może przejść do stanu zablokowania:

- Bezpośrednio ze stanu aktywności
- W wyniku wykonania funkcji exit (w Linuxie)
- Po wyczerpaniu kwantu czasu aktywności

\20.txt

Proces uśpiony (zablokowany):

- Zawsze usuwany jest z pamięci operacyjnej (kopiowany do obszaru wymiany)
- Oczekuje aż inny proces go zbudzi
- Oczekuje na zwolnienie procesora przez procesor aktywny

121.txt

Powłoka systemu operacyjnego:

- Określa zestaw dostępnych języków programowania
- Określa dostępny zestaw komend wewnętrznych
- Pozwala definiować zmienne środowiskowe, przekazywane do uruchamianych programów

122.txt

Pozycje katalogowe w systemie FAT:

- pozwalają definiować nazwy plików dowolnej wielkości
- zawierają informacje o położeniu pliku na dysku
- zawierają sumę kontrolną pliku

123.txt

Przydział ciągły miejsca na dysku:

- wymaga zapisu tablicy indeksowanej na dysku
- jest algorytmem optymalnym, nie dającym się zaimplementować w praktyce
- charakteryzuje się fragmentacją zewnętrzną przestrzeni dysku

124.txt

Planowanie zadań (scheduling) ma na celu:

- synchronizowanie dostępu do zasobów
- wybór procesu do aktywacji
- unikanie blokad

125.txt

Planowanie średnioterminowe:

- określa kolejność kopiowania procesów do pamięci z obszaru wymiany
- wybiera proces do uruchomienia spośród procesów gotowych
- określa procesy, które należy odblokować

126.txt

Sekcja krytyczna:

- To technika synchronizacji dostępu do pamięci współdzielonej przez procesor
- Wykorzystywana jest do negocjacji dostępu do semaforów

- Wykorzystywana jest od ochrony danych przed równoczesną ich modyfikacją przez procesy współbieżne

127.txt

Sygnały w systemie Linux:

- Generowane są wyłącznie przez sterowniki urządzeń zewnętrznych
- Nie mogą być ignorowane przez procesy
- Wysyłane są do pojedynczych procesów, grup procesów, procesów określonego użytkownika lub do wszystkich procesów

128.txt

Swobodny (zrandomizowany), dostęp do pliku:

- polega na współdzieleniu otwartych plików przez wiele procesów
- szereguje dostęp wielu procesów do pliku
- pozwala na występowanie plików nieciągłych, z niezaalokowanymi dziurami

129.txt

Stan procesu:

- przechowywany jest w jądrze systemu operacyjnego w obszarze uarea
- Identyfikuje procesy, które nie wymagają dostępu do urządzeń zewnętrznych
- określa, czy procesor jest przydzielony do procesu

130.txt

Stronicowanie na żądanie:

- wymaga tworzenia obrazu pamięci w obszarze wymiany
- wykorzystuje segmentację pamięci
- włączane jest na żądanie procesu (przez wywołanie funkcji swapon)

131.txt

Ścieżka do pliku:

- Ścieżka bezwzględna określa położenie pliku względem katalogu głównego:
- Może być względna, bezwzględna lub symboliczna
- Ścieżka symboliczna określa położenie pliku względem katalogu domowego użytkownika

132.txt

Warunek postępu w synchronizacji między procesami:

- To zapewnienie, że dwa procesy nie znajdą się w tym samym czasie w sekcji krytycznej
- To wymaganie, aby proces nie był usypiany przy wejściu do sekcji krytycznej, gdy nie ma w niej innych procesów
- Zapewnia, że proces nie jest wywłaszczany, gdy znajduje się w sekcji krytycznej

Warunek wykluczania w synchronizacji między procesami:

- Zapewnia, że proces nie jest wywłaszczany, gdy znajduje się w sekcji krytycznej
- To zapewnienie, że dwa procesy nie znajdują się w tym samym czasie w sekcji krytycznej
- Wymaga, aby proces miał dostęp do sekcji krytycznej, gdy nie przebywa w niej inny