

Politechnika Krakowska
im. Tadeusza Kościuszki

Wydział Fizyki, Matematyki i Informatyki

Systemy operacyjne

dr inż. Anna Jasińska-Suwada

Pula wszystkich pytań i zadań egzaminacyjnych
odpowiedzi

autorzy:
Starowicz Konrad
Starzak Mariusz
Śmierciak Aleksander

ostatnia aktualizacja: 7.02.2011, 18:53:51

Część teoretyczna

1. W jakich z metod przydziału dysku nie występuje problem fragmentacji zewnętrznej?
 - a) przydział ciągły,
 - b) mapa plików,
 - c) przydział listowy,
 - d) przydział indeksowy.
2. Podaj odpowiednią do systemów plikowych metodę przydziału miejsca:
FAT – tablica
ext2 – przydział indeksowy
3. Wybierz dwie metody przydziału miejsca na dysku, które najlepiej nadają się do realizacji dostępu bezpośredniego:
 - a) przydział ciągły,
 - b) mapa plików,
 - c) przydział listowy,
 - d) przydział indeksowy.
4. W jakich metodach zarządzania pamięcią oraz w jakich metodach przydziału miejsca na dysku występuje problem fragmentacji wewnętrznej?

W przypadku zarządzania pamięcią problem fragmentacji wewnętrznej występuje dla metody stronicowania.

W przypadku przydziału miejsca na dysku z kolei problem fragmentacji wewnętrznej występuje dla tablicy rozmieszczenia plików (FAT).

W przypadku zarządzania pamięcią problem fragmentacji zewnętrznej występuje dla metody segmentacji.

W przypadku przydziału miejsca na dysku z kolei problem fragmentacji zewnętrznej występuje dla metody przydziału ciągłego.

Niezagrożone fragmentacją zewnętrzną są przydziały listowe oraz indeksowe.
5. W których z metod przydziału miejsca na dysku występuje problem fragmentacji zewnętrznej?
 - a) przydział ciągły,
 - b) mapa plików,
 - c) przydział listowy,
 - d) przydział indeksowy.

6. Zakwalifikuj system plikowy FAT i ext2:

- | | |
|-------------------------------|-----------------------------|
| a) FAT – przydział ciągły, | ext2 – przydział indeksowy, |
| b) FAT – przydział listowy, | ext2 – przydział indeksowy, |
| c) FAT – mapa plików, | ext2 – przydział indeksowy, |
| d) FAT – przydział indeksowy, | ext2 – mapa plików. |

7. W systemie ext2 i-węzeł zawiera:

- a) typ pliku,
- b) prawa dostępu,
- c) liczbę dowiązań,
- d) nazwę pliku,
- e) adres pierwszego bloku danych,
- f) adres pośredniego bloku adresowego,
- g) 15-elementową tablicę z adresami.

8. Jakie informacje przechowywane są w katalogu w systemie plikowym FAT, a nie ma ich w katalogu w systemie ext2?

- a) adresy dyskowe wszystkich bloków danych ,
- b) nazwa pliku ,
- c) typ pliku ,
- d) rozmiar pliku ,
- e) adres tablicy FAT,
- f) numer pierwszego bloku danych.

Rozmiar pliku oraz typ pliku w systemie plikowym ext2 przechowywane są nie w plikach i katalogach, a w i-węzłach, do których wskaźniki mają pliki i katalogi.

9. Jakie informacje przechowywane są w katalogu w systemie plikowym FAT, a nie ma ich w katalogu w systemie ext3?

- a) adresy dyskowe wszystkich bloków danych,
- b) nazwa pliku,
- c) typ pliku,
- d) rozmiar pliku,
- e) adres tablicy FAT,
- f) numer pierwszego klastra danych.

10. W systemie ext3 i-węzeł pliku zawiera informacje:

- a) prawa dostępu do pliku,
- b) typ pliku,
- c) właściciel pliku,
- d) czas stworzenia, modyfikacji, dostępu oraz usunięcia pliku,
- e) rozmiar pliku w bajtach,
- f) liczba zajętych węzłów,
- g) tablica używana do zlokalizowania danych,
- h) flagi decydujące o zachowaniu się systemu przy dostępie do pliku określonego przez ten węzeł.

11. Jakie informacje przechowywane są w katalogu w ext3?

- a) adresy dyskowe,
- b) nazwa pliku,
- c) rozmiar pliku,
- d) numer węzła,
- e) typ pliku.

12. Jakie informacje przechowywane są w katalogu w FAT?

- a) nazwa pliku,
- b) typ pliku,
- c) rozmiar pliku,
- d) adres tablicy FAT,
- e) adres dyskowy pierwszego bloku danych,
- f) numer pierwszego bloku danych.

13. Jakie informacje przechowywane są w katalogu w NTFS?

- a) 64-bitowy odsyłacz do pliku,
- b) adres dyskowy pierwszego bloku danych,
- c) typ pliku,
- d) numer węzła,
- e) nazwa pliku,
- f) rozmiar pliku.

14. Wybierz właściwe cechy MFT w systemie plików NTFS:

- a) wszystkie atrybuty plików są rezydentne w MFT,
- b) metadane NTFS przechowywane są w plikach,
- c) metadane NTFS przechowywane są poza tablicą MFT,
- d) tablica MFT jest plikiem,
- e) MFT przechowuje wpisy na temat każdego pliku,
- f) każdy plik opisany jest co najmniej jednym rekordem w MFT,
- g) rozmiar rekordu jest parametrem systemu i wynosi <1-4 kB>,
- h) małe atrybuty plików są rezydentne w MFT,
- i) katalogi w NTFS mają strukturę B-drzewa,
- j) każdy plik ma 64-bitowy identyfikator,
- k) każdy plik ma 16-bitowy identyfikator,
- l) każdy plik ma 48-bitowy identyfikator,
- m) katalogi w NTFS mają strukturę listy dwukierunkowej.

15. Co to jest superblok w systemie plików ext3, gdzie się znajduje i jakie informacje zawiera?

Superblok zawiera metadane dotyczące systemu plików, dla każdego systemu plików istnieje jeden superblok umieszczony na początku systemu plików na dysku. Jądro odczytuje superblok podczas montowania systemu plików i trzyma go w pamięci operacyjnej aż do chwili zdemontowania systemu plików.

Superblok przechowuje następujące informacje:

- 1) rozmiar systemu plików w blokach,
- 2) rozmiar listy i-węzłów w blokach,
- 3) liczbę wolnych bloków oraz wolnych i-węzłów,
- 4) listę wolnych bloków i listę wolnych i-węzłów.

16. W bieżącym katalogu są pliki:

ko1.aaa ko?.qww ko\ a ko?.o ko?a.a
Wydanie polecenia `ls ko\?.*` spowoduje wyświetlenie się nazw plików `ko?.qww` i `ko?.o`.

17. Aby usunąć plik pl1 z katalogu kat1 potrzebne są następujące prawa:

- a) w dla katalogu kat1,
- b) w dla pliku pl1,
- c) r dla pliku pl1,
- d) x dla katalogu kat1,
- e) x dla pliku pl1,
- f) r dla katalogu kat1.

18. W katalogu zoo jest katalog zwierz. Aby uruchomić plik smok z katalogu zwierz konieczne są następujące prawa:

zoo: --x
zwierz: --x
smok: --x

19. W katalogu zoo jest katalog zwierz. Aby wylistować plik smok z katalogu zwierz konieczne są następujące prawa:

zoo: --x
zwierz: r-x
smok: ---

20. Które z poniższych zdań jest prawdziwe?

- a) link twardy może wskazywać tylko na plik zwykły,
- b) link symboliczny może wskazywać na katalog lub plik,
- c) plik oraz link symboliczny do niego mają taki sam rozmiar,
- d) plik oraz link twardy do niego mają ten sam numer węzła.

21. Jeśli plik ma ustawione prawa dostępu: **rwsr--r-x** to prawo wykonywania pliku ma:

właściciel oraz inni

Takie prawa można nadać poleceniem (**chmod** numeryczne):

chmod 4745 plik

22. Jeśli plik ma ustawione prawa dostępu: **rwS-----x** to prawo wykonywania pliku ma:

inni

Takie prawa można nadać poleceniem (**chmod** numeryczne):

chmod 4601 plik

23. Jeśli plik ma ustawione prawa dostępu: **rwSr-----x**, to prawo wykonywania pliku ma:

a) właściciel pliku,

b) członek grupy,

c) pozostali użytkownicy systemu.

Takie prawa można nadać poleceniem (**chmod** numeryczne):

chmod 4641 plik

24. Na czym polega różnica między blokowaniem a nieblokowaniem wywołaniem operacji wejścia/wyjścia?

Blokowane wywołanie systemowe powoduje wstrzymanie procesu wywołującego (umieszczenie w kolejce procesów czekających), natomiast nieblokowane procesu owego nie wstrzymuje; zostaje on szybko zakończony, zwracając liczbę przesłanych bajtów (niekoniecznie wszystkich).

25. Instrukcje występujące w kodzie programu po wywołaniu funkcji **exec** wykonywane są przez:

a) proces potomny,

b) proces rodzica,

c) proces potomny i rodzica,

d) żaden z nich.

26. Funkcja **fork** zwraca w procesie potomnym:

zero,

a w procesie macierzystym:

PID procesu potomnego.

27. Komentarze w powłoce **BASH** zaczynają się od znaku/znaków:

a) **//**

b) **!**

c) **#**

d) **\$**

28. Jak w skrypcie BASH odwołać się do identyfikatora (PID) procesu?

- a) \$1
- b) \$(1)
- c) \$0
- d) \$#
- e) \$\$

29. Jak w skrypcie BASH odwołać się do pierwszego argumentu z jakim go wywołano?

- a) \$1
- b) \$(1)
- c) \$0
- d) \$#
- e) \$\$

30. Jakie polecenia mogą spowodować zmianę liczby dowiązań do pliku?

- a) ln
- b) mv
- c) ln -s
- d) rm

31. Inkrementacji zmiennej i w skrypcie BASH można dokonać poleceniem:

- a) i=\$((i+1))
- b) i=\${i+1}
- c) let i=\$i +1
- d) let i=\$i+1
- e) i=i++
- f) i+=1

32. Inkrementacji zmiennej i w skrypcie TCSH można dokonać poleceniem:

- a) i=\$((i+1))
- b) set i=\$i+1
- c) let i=\$i +1
- d) @ i=i++
- e) @ i++

33. Aby zdefiniować nową zmienną lokalną PIES i nadać jej wartość jamnik w powłoce BASH należy wykonać operację:

- a) set PIES jamnik
- b) PIES=jamnik
- c) expr PIES jamnik
- d) PIES:=jamnik

34. Aby ustawić zmienną PIES na wartość jamnik w powłoce TCSH należy wykonać operację:

- a) `set PIES = jamnik`
- b) `jamnik=PIES`
- c) `PIES=jamnik`
- d) `let PIES=jamnik`

35. Zmienna specjalna powłoki BASH która zawiera listę katalogów rozdzielonych dwukropkami, które są przeszukiwane w celu znalezienia polecenia do uruchomienia to:

- a) HOME
- b) CDPATH
- c) PATH
- d) IFS

36. Polecenie z historii nr 3 w powłoce BASH wykonuje się:

- a) `!3`
- b) `3`
- c) `history 3`
- d) `./bash_history 3`

37. Wybierz poprawne definicje aliasu powłoki BASH:

- a) `alias l=ls`
- b) `alias ls2="ls -al"`
- c) `alias ls3='ls -al \! *'`
- d) `alias ls=`ls -al``
- e) `alias ls1 'ls -la'`

Należy zwrócić uwagę na symbole, które są podobne do znaków apostrofu, ale nimi nie są.

38. Jak w skrypcie BASH wczytać zmienną ZM?

- a) `> ZM`
- b) `>> ZM`
- c) `set ZM`
- d) `read ZM`

39. Jak można przekazać alias do innej powłoki?

- a) nie można,
- b) poleceniem `export (bash)`,
- c) umieszczając go w pliku startowym logowania,
- d) umieszczając go w pliku `.bashrc`,
- e) umieszczając go w pliku `.bash_profile`.

40. Jak można w powłoce TCSH „przekazać” alias do innej powłoki?

Poprzez wpisanie aliasu do pliku `.tcshrc`.

41. Rozszerzone prawo dostępu do pliku (s zamiast x) dla użytkownika oznacza, że:

- a) tylko właściciel ma prawo wykonywania pliku,
- b) proces wykonujący ten plik otrzymuje prawa właściciela pliku,
- c) ten plik może usunąć tylko właściciel,
- d) liczby "nice" dla procesu wykonującego ten plik nie można zmienić.

42. Funkcje związane z przekształcaniem czasu to:

1. mktime 2. gmtime 3. localtime 4. strftime 5. strptime

Wybierz właściwe funkcje (1-5) dla zamiany struktur określających czas:

- | | | | | | | | |
|----------------|-----|-------------|---|---|---|---|---|
| a) struct tm | - > | time_t | 1 | 2 | 3 | 4 | 5 |
| b) time_t | - > | struct tm | 1 | 2 | 3 | 4 | 5 |
| c) struct tm | - > | ciąg znaków | 1 | 2 | 3 | 4 | 5 |
| d) ciąg znaków | - > | struct tm | 1 | 2 | 3 | 4 | 5 |

43. Aby posortować linie pliku lista, ponumerować je, wynik zapisać w pliku nowalista oraz wyświetlić na monitorze należy użyć polecenia:

- a) cat -n lista | sort | tee nowalista
- b) sort lista | cat -n > nowalista
- c) sort lista | cat -n | tee nowalista
- d) sort | cat -n lista | tee nowalista

44. Funkcja open:

- a) tworzy ścieżkę dostępu do pliku lub urządzenia,
- b) zwraca niepowtarzalny deskryptor pliku,
- c) nadaje prawa dostępu zależne od umask,
- d) zwraca najniższy nieużywany deskryptor pliku,
- e) nadaje prawa dostępu do pliku zależne jedynie od parametrów wywołania.

45. Funkcja dup:

- a) umożliwia czytanie i pisanie w różnych miejscach pliku,
- b) zwraca nowy deskryptor i zamyka stary deskryptor,
- c) zwraca najniższy wolny deskryptor,
- d) umożliwia komunikację pomiędzy procesami spokrewnionymi.

46. Funkcja popen:

- a) umożliwia przekazywanie danych pomiędzy spokrewnionymi procesami,
- b) wywołuje nową powłokę,
- c) zwraca deskryptor pliku,
- d) jest funkcją wysokopoziomową.

47. Funkcja pipe:

- a) tworzy potok nazwany,
- b) umożliwia komunikację pomiędzy dowolnymi procesami,
- c) zwraca parę strumieni plikowych,
- d) zwraca dwa deskryptory plików.

Funkcja `pipe` nie zwraca dwóch deskryptorów plików. Funkcja `pipe` tworzy parę deskryptorów plików, które wskazują na i-węzeł potoku, i umieszcza je w tablicy wskazywanej przez podany jej parametr typu `int tablica [2]`.

Deskryptor pod wskaźnikiem `tablica [0]` służy do czytania, a deskryptor pod wskaźnikiem `tablica [1]` do pisania.

48. Funkcja mkfifo:

- a) tworzy potok nazwany,
- b) umożliwia komunikację pomiędzy dowolnymi procesami,
- c) zwraca parę strumieni plikowych,
- d) zwraca dwa deskryptory plików.

49. Polecenie `tar czvf arch pr`:

- a) jest błędne,
- b) tworzy archiwum o nazwie `arch.tar` z plikami z katalogu `pr`,
- c) tworzy archiwum o nazwie `arch`, do którego będzie można dodawać pliki,
- d) tworzy archiwum, którego nie da się uaktualnić,
- e) wyświetla nazwy plików wchodzących w skład archiwum.

50. Co to jest deadlock?

- a) blokada pamięci,
- b) stan, w którym procesy są zawieszone oczekując na niemożliwe zdarzenie,
- c) stan, w którym procesy jednocześnie zmieniają wartość wspólnych danych,
- d) błąd w dostępie do strony pamięci wirtualnej.

51. Przyporządkuj poziomy RAID do kategorii: 1- paskowanie, 2 - zbiór pasków z parzystością (na kolejnych dyskach), 3 - dyski lustrzane, 4 - zbiór pasków z parzystością (na jednym dysku)

Poziom 0	-	1- paskowanie
Poziom 1	-	3 - dyski lustrzane
Poziom 3	-	4 - zbiór pasków z parzystością (na jednym dysku)
Poziom 5	-	2 - zbiór pasków z parzystością (na kolejnych dyskach)

52. Instrukcje występujące w kodzie programu po wywołaniu funkcji `fork` wykonywane są przez:

- a) proces potomny,
- b) proces rodzica,
- c) proces potomny i rodzica,
- d) żaden z nich.

53. Które ze zdań opisujących organizację pamięci wirtualnej są prawdziwe:

- a) w przypadku stronicowania pamięć rzeczywista podzielona jest na równe bloki,
- b) w przypadku stronicowania podział adresu na numer strony i bajtu jest wykonywany sprzętowo,
- c) wielosegmentowa pamięć wirtualna dzieli się na spójne logicznie, równe segmenty,
- d) segmentacja może prowadzić do fragmentacji zewnętrznej,
- e) segmentacja prowadzi do fragmentacji wewnętrznej,
- f) odwrócona tablica stron zawiera informacje o identyfikatorach procesów,
- g) adres wirtualny może składać się z numerów: segmentu, strony, bloku.
- h) tablica stron jest wspólna dla wszystkich procesów.

54. Macierz dostępu w systemie Linux zorganizowana jest za pomocą:

- a) wykazu dostępu do obiektów,
- b) tablicy globalnej,
- c) wykazy uprawnień do domen,
- d) mechanizmu zamka-klucza.

55. Jaka liczba reprezentuje deskryptor standardowego wyjścia w systemie Linux?

- a) 0
- b) 1
- c) STDOUT_FILENO
- d) 2

stdout - standardowe wyjście (deskryptor 1)	- przeważnie monitor,
stderr - standardowe wyjście błędów (deskryptor 2)	- przeważnie monitor,
stdin - standardowe wejście (deskryptor 0)	- przeważnie klawiatura.

56. Pamięć podręczna:

- a) w hierarchii pamięci leży powyżej pamięci operacyjnej (pamięć dyskowa jest poniżej),
- b) przechowuje informacje przejściowo,
- c) problem zgodności pamięci podręcznej dotyczy środowiska wieloprocessorowego,
- d) ponad 80% dostępu powinno się do niej odwoływać.

57. Kategorie „czystych systemów operacyjnych”:

- a) offline,
- b) wieloużytkownikowy,
- c) online,
- d) real-time,
- e) system wielozadaniowy.

58. Algorytm bankiera:

- a) służy do wykrywania blokad,
- b) zapobiega blokadom,
- c) jest wywoływany gdy zamówienie zasobu nie może być zrealizowane,
- d) wymaga zadeklarowania liczby zasobów dla wszystkich procesów.

59. Algorytm wykrywania zakleszczeń:

- a) to algorytm bankiera,
- b) umożliwia wykrycie „sprawcy” zakleszczenia,
- c) bywa wywoływany gdy zamówienie zasobu nie może być zrealizowane,
- d) wymaga zadeklarowania liczby zasobów dla wszystkich procesów.

60. Niewywłaszczające algorytmy szeregowania zadań to:

- a) FCFS,
- b) SJF,
- c) priorytetowy,
- d) rotacyjny (round robin).

SRTF (ang. Shortest-Remaining-Time-First), czyli "najpierw zadanie o najkrótszym pozostałym czasie wykonania" to wersja strategii SJF z wywłaszczaniem.

Algorytm priorytetowy może być wywłaszczający jak i niewywłaszczający.

61. Wywłaszczające algorytmy szeregowania zadań to:

- a) FCFS,
- b) SJF,
- c) priorytetowy,
- d) rotacyjny (round robin).

62. Wymień algorytmy elekcji (wyboru koordynatora w systemach rozproszonych).

Algorytm tyrana i algorytm pierścieniowy.

63. Do IPC – mechanizmów komunikacji międzyprocesowej należą:

- a) monitory,
- b) semaforey,
- c) wyrażenia ścieżkowe,
- d) komunikaty,
- e) kolejki FIFO,
- f) rejony krytyczne.

64. Jaka funkcja zwraca najniższy wolny deskryptor pliku?

- a) open
- b) dup
- c) dup2
- d) popen

65. Jaka funkcja zwraca parę deskryptorów pliku?

- a) pipe
- b) mkfifo
- c) popen
- d) open
- e) dup2

Funkcja `pipe` nie zwraca dwóch deskryptorów plików. Funkcja `pipe` tworzy parę deskryptorów plików, które wskazują na i-węzeł potoku, i umieszcza je w tablicy wskazywanej przez podany jej parametr typu `int` tablica [2].

Deskryptor pod wskaźnikiem `tablica [0]` służy do czytania, a deskryptor pod wskaźnikiem `tablica [1]` do pisania.

66. Funkcja `semget`:

- a) tworzy nowy semafor,
- b) zwraca klucz istniejącego semafora,
- c) służy do zmiany wartości semafora,
- d) zwraca wskaźnik do tablicy struktur `sembuf`,
- e) usuwa semafor .

67. Funkcja `msgget`:

- a) tworzy nowy semafor,
- b) zwraca identyfikator istniejącej kolejki komunikatów,
- c) służy do wysyłania komunikatów,
- d) zwraca wskaźnik do struktury `msg`,
- e) usuwa kolejkę.

68. Klucz generowany np. przez funkcję `ftok` może być używany do:

- a) tworzenia kolejki komunikatów lub uzyskania dostępu do niej – `msgget`,
- b) uzyskania dostępu do zestawu semaforów -`semget`,
- c) tworzenia segmentu pamięci dzielonej – `shmget`,
- d) wysłania/odbioru komunikatu - `msgsnd`, `msgrcv`,
- e) wykonania operacji kontrolnych na semaforze – `semctl`,
- f) tworzenia - zestawu semaforów – `semget`.

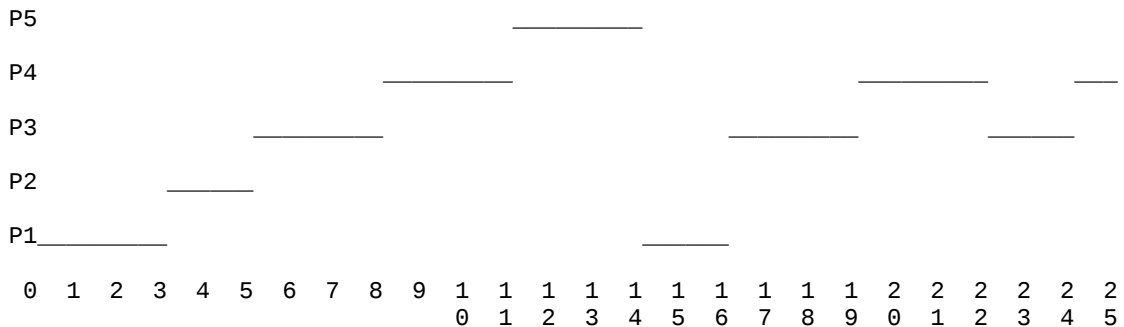
Część zadaniowa

1. Dla procesów o czasach trwania faz wynoszących kolejno: **P1 - 5, P2 - 2, P3 - 8, P4 - 7, P5 - 3** policz średni czas oczekiwania i narysuj grafy czasowe dla algorytmów szeregowania zadań:

- 1) rotacyjnego (z kwantem czasu wynoszącym 3 jednostki),
 - 2) **SJF** z wywłaszczeniem, przy założeniu, że czasy przybycia procesów wynoszą odpowiednio: **P1 - 0, P2 - 2, P3 - 4, P4 - 6, P5 - 7** jednostek.
- 1) Dla algorytmu rotacyjnego (round-robin):

Kwant czasu wynosi 3 jednostki

Proces	P1	P2	P3	P4	P5
Czas trwania fazy procesu [ms]	5	2	8	7	3



P1 - 11ms

P2 - 3ms

P3 - 16ms

P4 - 18ms

P5 - 11ms

Sumujemy:

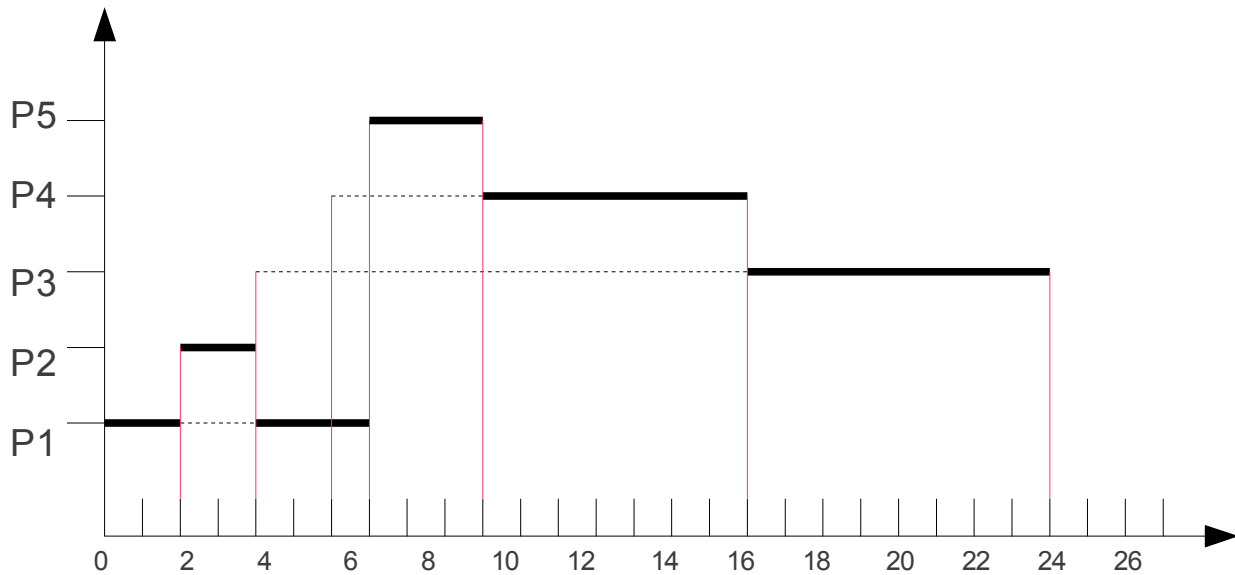
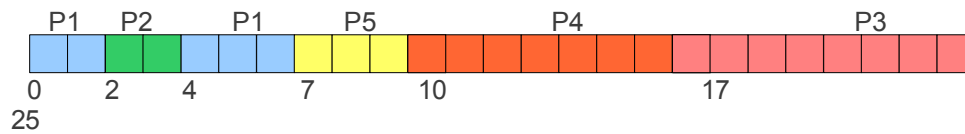
$$(11 + 3 + 16 + 18 + 11) = 59 \text{ ms}$$

$$\frac{59}{5} = 11,8 \text{ ms}$$

Średni czas oczekiwania dla procesów wynosi 11,8ms.

2) Dla algorytmu SJF z wywłaszczeniem:

Proces	P1	P2	P3	P4	P5
Czas przybycia procesu	0	2	4	6	7
Czas trwania fazy procesu	5	2	8	7	3



Sumujemy wszystkie oczekiwania procesu na wznowienie (przerywane linie na wykresie):

$$(2+0+13+4+0)=19ms$$

$$\frac{19}{5}=3,8ms$$

Średni czas oczekiwania dla procesów wynosi 3,8ms.

2. **Zaproponuj rozwiązanie problemu pięciu filozofów z wykorzystaniem wybranych mechanizmów komunikacji między procesami – przedstaw pseudokod.**

```
var
    paleczki: array [0..4] of semaphore;
    talerze: semaphore;

procedure filozof (i: 0..4);
begin
    repeat
        myśl;
        P (talerz);
        P (paleczki [i]);
        P (paleczki [(i + 1) mod 5]);
        jedz;
        V (paleczki [i]);
        V (paleczki [(i + 1) mod 5]);
        V (talerz);
    until false
end;
```

3. **Zaproponuj rozwiązanie problemu producent-konsument z wykorzystaniem wybranych mechanizmów komunikacji między procesami - przedstaw pseudokod.**

```
semaphore pełny = 0
semaphore pusty = ROZMIAR_BUFORA

procedure producent ()
{
    while (true)
    {
        produkt = produkuj ()
        down (pusty)
        dodajProduktDoBufora (produkt)
        up (pełny)
    }
}

procedure konsument ()
{
    while (true)
    {
        down (pełny)
        produkt = pobierzProduktZBufora ()
        up (pusty)
        użyjProdukt (produkt)
    }
}
```


4. Zsynchronizuj trzy procesy: **P1**, **P2**, **P3** przy pomocy semaforów (podaj pseudokody procesów). Proces **P1** składa się z podzadań: **a1**, **b1** i **c1**, proces **P2** składa się z podzadań: **a2** i **b2**, proces **P3** składa się z podzadań: **a3**, **b3** i **c3**. Dana jest relacja pierwszeństwa w postaci zbioru par: $\{ (a2, b1), (a3, b2), (c1, a2), (b2, c2) \}$. Przy założeniu, że każde z podzadań procesu **P1** trwa 1 jednostkę, procesu **P2** 2 jednostki, a procesu **P3** 3 jednostki czasu narysuj wykres czasowy procesów, zaznacz potencjalnie współbieżne podzadania i podaj jaki jest minimalny czas wykonania pojedynczej serii zadań (**P1**, **P2** i **P3**) i przy jakich założeniach.

5. Sprawdź, czy system jest w stanie bezpiecznym i ewentualnie podaj bezpieczny ciąg procesów.

Procesy: **P1, P2, P3**

Zasoby: **Z1, Z2, Z3**

Z1 - 6 egzemplarzy

Z2 - 8 egzemplarzy

Z3 - 6 egzemplarzy

Maksymalne deklaracje procesów:

P1: Z1 - 4, Z2 - 7, Z3 - 5

P2: Z1 - 3, Z2 - 4, Z3 - 4

P3: Z1 - 2, Z2 - 5, Z3 - 4

Zaalokowane zasoby:

P1: Z1 - 2, Z2 - 2, Z3 - 0

P2: Z1 - 1, Z2 - 4, Z3 - 4

P3: Z1 - 1, Z2 - 2, Z3 - 0

System jest w stanie bezpiecznym, jeśli istnieje ciąg procesów, który jest bezpieczny. Bardziej obrazowo: system jest w stanie bezpiecznym, jeśli istnieje taki ciąg procesów, że nie dochodzi do zakleszczenia – żaden proces z owego ciągu nie ma maksymalnej deklaracji zasobów większej niż aktualnie dostępne zasoby.

Dostępne zasoby = [łączna liczba egzemplarzy] - [suma w danym Zk wszystkich procesów z kolumny Zaalokowane zasoby]

$$Z1 = 6 - (2 + 1 + 1) = 2$$

$$Z2 = 8 - (2 + 4 + 2) = 0$$

$$Z3 = 6 - (0 + 4 + 0) = 2$$

	Zaalokowane zasoby			Maksymalne deklaracje			Dostępne zasoby		
	Z1	Z2	Z3	Z1	Z2	Z3	Z1	Z2	Z3
P1	2	2	0	4	7	5	2	0	2
P2	1	4	4	3	4	4			
P3	1	2	0	2	5	4			

Szukamy procesu, który może się wykonać jako pierwszy. Każdy proces po wykonaniu zwalnia wcześniej zajęte zasoby, zwiększając nasze możliwości znajdowania kolejnych elementów bezpiecznego ciągu.

Spróbujmy P1:

Z1: ma go 2, maksymalnie potrzebuje 4, dostępnych 2 – OK

Z2: ma go 2, maksymalnie potrzebuje 7, dostępnych 0 – ŻLE

Spróbujmy P2:

Z1: ma go 1, maksymalnie potrzebuje 3, dostępnych 2 – OK

Z2: ma go 4, maksymalnie potrzebuje 4, dostępnych 0 – OK

Z3: ma go 4, maksymalnie potrzebuje 4, dostępnych 2 – OK

A więc hipotetycznie pierwszym elementem bezpiecznego ciągu jest proces P2. Szukamy dalej, pamiętając o zaktualizowaniu dla nowego stanu kolumny Dostępne zasoby w tabeli:

Dostępne zasoby		
Z1	Z2	Z3
$2 + 1 = 3$	$0 + 4 = 4$	$2 + 4 = 6$

Spróbujmy P1:

Z1: ma go 2, maksymalnie potrzebuje 4, dostępnych 3 – OK

Z2: ma go 2, maksymalnie potrzebuje 7, dostępnych 4 – ŻŁE

Spróbujmy P3:

Z1: ma go 1, maksymalnie potrzebuje 2, dostępnych 3 – OK

Z2: ma go 2, maksymalnie potrzebuje 5, dostępnych 4 – OK

Z3: ma go 0, maksymalnie potrzebuje 4, dostępnych 6 – OK

A więc hipotetycznie drugim elementem bezpiecznego ciągu jest proces P3. Sprawdzamy, czy ciąg ten rzeczywiście jest ciągiem bezpiecznym przez upewnienie się, że ostatnim (trzecim) elementem tego ciągu jest proces P1, po uprzednim zmodyfikowaniu tablicy:

Dostępne zasoby		
Z1	Z2	Z3
$3 + 1 = 4$	$4 + 2 = 6$	$6 + 0 = 6$

Spróbujmy P1:

Z1: ma go 2, maksymalnie potrzebuje 4, dostępnych 4 – OK

Z2: ma go 2, maksymalnie potrzebuje 7, dostępnych 6 – OK

Z3: ma go 0, maksymalnie potrzebuje 5, dostępnych 6 – OK

Tak więc system jest w stanie bezpiecznym, gdyż istnieje ciąg, procesów, który jest bezpieczny. Ciągiem ten jest ciąg $P2 \rightarrow P3 \rightarrow P1$.

6. Ile komunikatów typu ELEKCJA musi maksymalnie zostać wysłanych w celu wyboru nowego koordynatora w przypadku zastosowania algorytmu tyrana dla n procesów przy założeniu, że tylko jeden proces zauważył brak koordynatora?

a) $n + n-1 + n-2 + \dots + 2$

b) $n-1 + n-2 + n-3 + \dots + 1$

c) $2(n-1)$

d) $n + n-1 + n-2 + \dots + 1$

n – liczba procesów

p – liczba procesów, które zauważyły brak koordynatora

$ile = (n-p) + (n-p-1) + (n-p-2) + \dots + 1$

$p=1 \Rightarrow ile = (n-1) + (n-2) + (n-3) + \dots + 1$

7. **Policz średni czas oczekiwania dla algorytmu szeregowania zadań SJF dla procesów o czasach trwania faz wynoszących kolejno: 5, 3, 2, 4, 6 jednostek. Czy jest to algorytm wywłaszczający czy nie?**

1) Układamy procesy w kolejności SJF: 2 3 4 5 6

2) Obliczenia:

$$(0+2+5+9+14)=30\text{ ms}$$

$$\frac{30}{5}=6\text{ ms}$$

3) Średni czas oczekiwania w tym przypadku 6 jednostek czasu.

Jest to algorytm niewywłaszczający.

8. **Oblicz liczbę braków stron dla algorytmu LRU dla 3 i 4 ramek, dla zredukowanego ciągu odwołań do pamięci postaci: 5 3 2 5 3 1 4 2 5 3 4 1 3.**

Dla 3 ramek – 10 braków stron.

Lp.	1	2	3	4	5	6	7	8	9	10	11	12	13
Odwołanie	5	3	2	5	3	1	4	2	5	3	4	1	3
R1	5	5	5	5	5	5	4	4	4	3	3	3	3
R2		3	3	3	3	3	3	2	2	2	4	4	4
R3			2	2	2	1	1	1	5	5	5	1	1

Dla 4 ramek – 9 braków stron.

Lp.	1	2	3	4	5	6	7	8	9	10	11	12	13
Odwołanie	5	3	2	5	3	1	4	2	5	3	4	1	3
R1	5	5	5	5	5	5	5	2	2	2	2	1	1
R2		3	3	3	3	3	3	3	5	5	5	5	5
R3			2	2	2	2	4	4	4	4	4	4	4
R4						1	1	1	1	3	3	3	3

9. Oblicz liczbę braków stron dla algorytmu optymalnego dla 3 i 4 ramek, dla zredukowanego ciągu odwołań do pamięci postaci: **5 3 2 5 3 1 4 2 5 2 4 5 3**.

Dla 3 ramek – 6 braków stron.

Lp.	1	2	3	4	5	6	7	8	9	10	11	12	13
Odwołanie	5	3	2	5	3	1	4	2	5	2	4	5	3
R1	5	5	5	5	5	5	5	5	5	5	5	5	3
R2		3	3	3	3	1	4	4	4	4	4	4	4
R3			2	2	2	2	2	2	2	2	2	2	2

Dla 4 ramek – 5 braków stron.

Lp.	1	2	3	4	5	6	7	8	9	10	11	12	13
Odwołanie	5	3	2	5	3	1	4	2	5	2	4	5	3
R1	5	5	5	5	5	5	5	5	5	5	5	5	5
R2		3	3	3	3	3	3	3	3	3	3	3	3
R3			2	2	2	2	2	2	2	2	2	2	2
R4						1	4	4	4	4	4	4	4

10. Oblicz liczbę braków stron dla algorytmu **FCFS** oraz optymalnego dla 3 ramek, dla zredukowanego ciągu odwołań do pamięci postaci: **3 2 1 5 3 2 5 3 4 1 3**.

Dla algorytmu FCFS – 9 braków stron.

Lp.	1	2	3	4	5	6	7	8	9	10	11
Odwołanie	3	2	1	5	3	2	5	3	4	1	3
R1	3	3	3	5	5	5	5	5	4	4	4
R2		2	2	2	3	3	3	3	3	1	1
R3			1	1	1	2	2	2	2	2	3

Dla algorytmu optymalnego – 6 braków stron.

Lp.	1	2	3	4	5	6	7	8	9	10	11
Odwołanie	3	2	1	5	3	2	5	3	4	1	3
R1	3	3	3	3	3	3	3	3	3	3	3
R2		2	2	2	2	2	2	2	4	4	4
R3			1	5	5	5	5	5	5	1	1

11. **Implementacja operacji semaforowej wait (z obsługą kolejek procesora).**

12. **Opisy katalogów domowych użytkownika janek i marek:**

d rwx----- janek orly janek

d rwx----- marek orly marek

Opis pliku wykonaj z katalogu domowego użytkownika janek :

. rwx----- janek orly wykonaj

Napisz konieczne do wykonania przez janka polecenia, aby marek mógł uruchomić program wykonaj i aby proces ten działał z takimi prawami, jakie ma janek.

setfacl -m u:marek:rwx orly

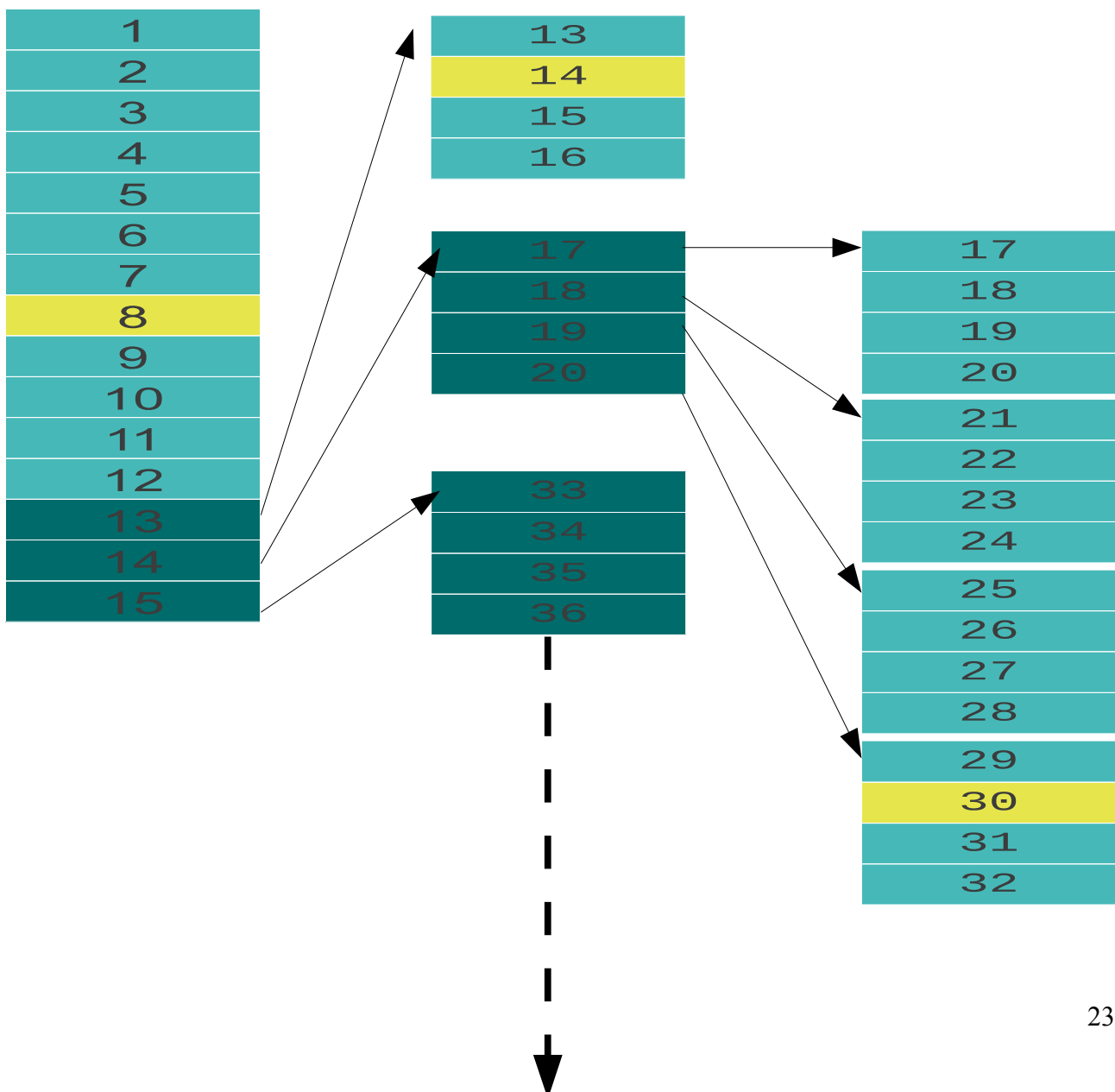
setfacl -m u:marek:rws orly/wykonaj

13. Rozważ system plikowy ext2. Rozmiar bloku wynosi 16B. Plik zajmuje 60 bloków dyskowych. Zaznacz na rysunku gdzie znajdują się adresy bloków pliku o numerach porządkowych: 8, 14, 30.

Na potrzeby tłumaczenia wprowadzam uproszczone nazewnictwo: blok jest strukturą, która ma swoje składniki – są nimi elementy (adresy bloków pliku) lub wskaźniki (do innych bloków).

Rozmiar bloku wynosi $n=16B$, a więc mamy 15-składnikowy blok „główny”, w którym:

- pod numerami 1. - 12. znajdują się pojedyncze elementy,
- pod numerem 13. znajduje się wskaźnik do 4-elementowego bloku (gdyż $(\frac{n}{4})^1=(\frac{16}{4})^1=4[B]$); numery: 13. -16. ,
- pod numerem 14. znajduje się wskaźnik do 4-składnikowego bloku, w którym składnikami są wskaźniki do 4-elementowych bloków (gdyż $(\frac{n}{4})^2=(\frac{16}{4})^2=16[B]$); numery: 17. - 32. ,
- pod numerem 15. znajduje się wskaźnik do 4-składnikowego bloku, w którym składnikami są wskaźniki do 4-składnikowych bloków, w których składnikami są wskaźniki do 4-elementowych bloków (gdyż $(\frac{n}{4})^3=(\frac{16}{4})^3=64[B]$); numery: 33. - 64. .



14. Oblicz maksymalną liczbę bloków dyskowych w systemie ext3 dostępną dla jednego pliku, jeśli rozmiar bloku wynosiłby 64B.

W systemach plikowych ext2 i ext3 maksymalny rozmiar pliku jest ograniczony przez

$$\min \left(\left(\left(\frac{b}{4} \right)^3 + \left(\frac{b}{4} \right)^2 + \frac{b}{4} + 12 \right) * b, 2^{41} \right), \text{ gdzie } b \text{ jest rozmiarem bloku.}$$

Maksymalną liczbę bloków dyskowych dostępną dla jednego pliku otrzymamy dzieląc maksymalny rozmiar pliku przez rozmiar bloku.

Podstawiając pod $b = 64B$ otrzymujemy wartość 280320, a następnie dzieląc przez 64 – wartość 4380, która to jest maksymalną liczbą bloków dyskowych, dostępną dla jednego pliku.

15. Przeanalizuj fragment kodu i odpowiedz ile procesów zostanie utworzonych, ile razy pojawi się na ekranie litera A, B, C, D oraz podaj przykładową poprawną sekwencję ich wystąpienia. Naskicuj drzewo procesów.

```
fork ();
printf ("B\n");
fork ();
if (fork()==0)
printf („C\n");
execvp ("echo","echo","A", NULL);
fork ();
printf ("D\n");
```

W sumie utworzonych zostanie osiem procesów.

Przykładowa sekwencja: B C A A B A A C A C A C A A

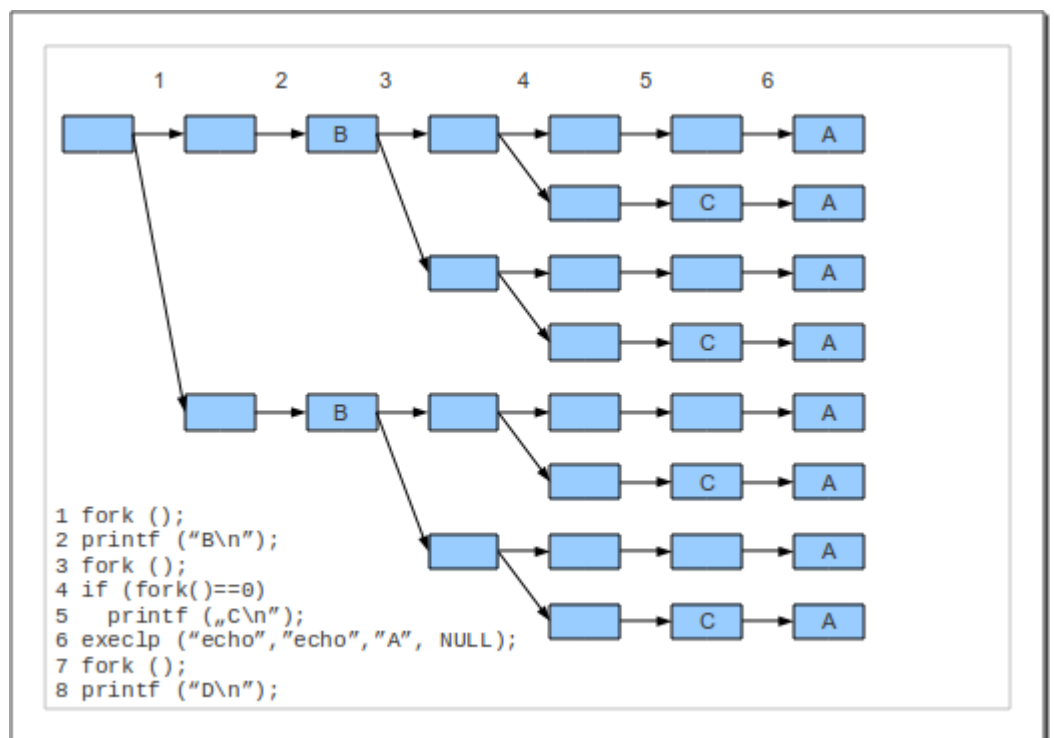
Wystąpienia:

A - 8 razy

B - 2 razy

C - 4 razy

D - 0 razy



16. Masz do dyspozycji linie kodu:

1) if (fork () == 0)

2) exit (0)

3) sleep (40)

4) wait (NULL)

Zbuduj za pomocą nich fragment kodu tworzący proces sierotę i proces zombie (używaj numerów 1-4 i dodatkowo nawiasów).

Program tworzący proces – sierotę:

```
if (fork() == 0)
{
    sleep(40);
    exit(0);
}
exit(0);
```

Zapis numerowy, zgodnie z poleceniem:

```
1
{
    3;
    2;
}
2;
```

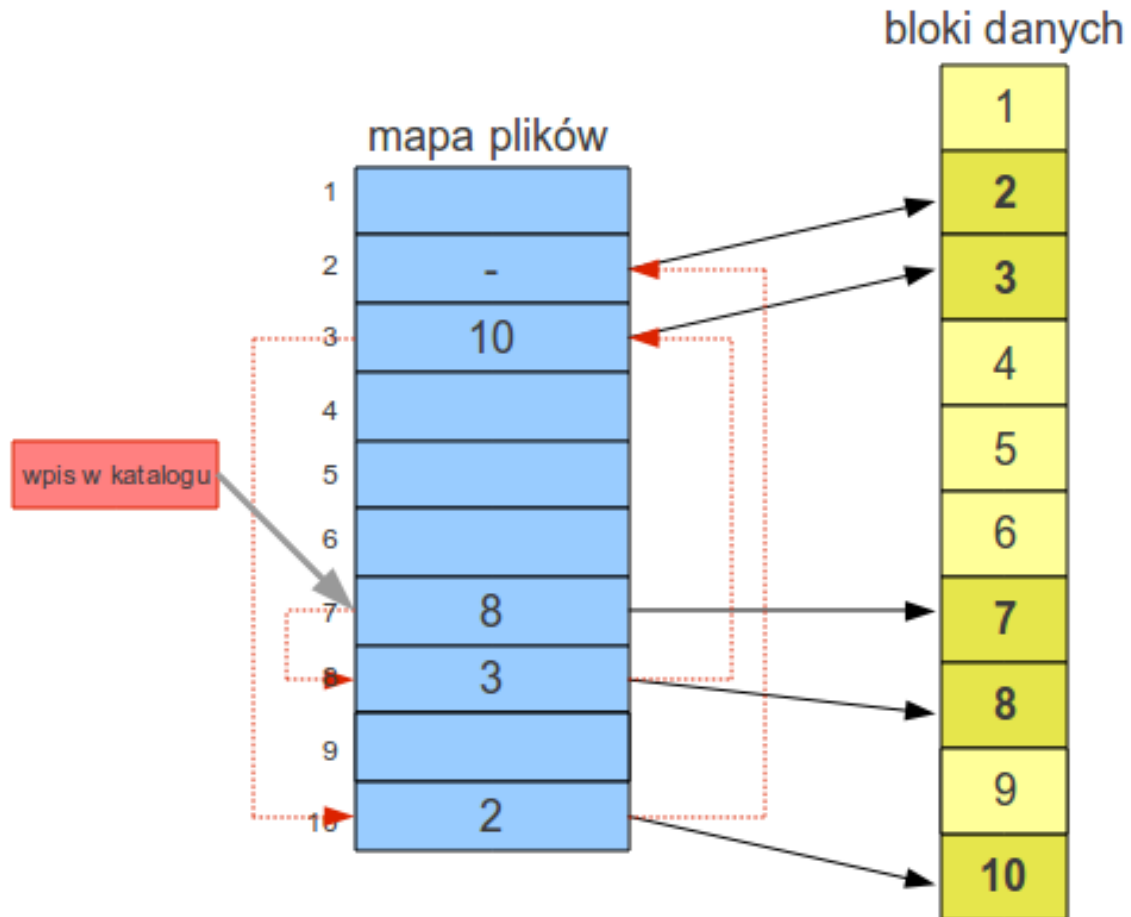
Program tworzący proces – zombie:

```
if (fork() == 0)
{
    exit(0);
}
sleep(40);
wait(NULL);
```

Zapis numerowy, zgodnie z poleceniem:

```
1
{
    2;
}
3;
4;
```

17. Rozważ metodę przydziału miejsca na dysku: mapę plików. Plik o nazwie dane zajmuje bloki na dysku: 7, 8, 3, 10, 2. Wypełnij informacje o tym pliku w katalogu, mapie plików oraz zaznacz bloki zajęte przez ten plik w obszarze danych na dysku.



18. Przeanalizuj poniższy program. Ile procesów zostanie uruchomionych w wyniku jego wykonania, ile linii : pid = ... pojawi się na ekranie dla obu przypadków: a i b?

```
a) int main ()
{
    fork ();
    if (fork () == 0)
    {
        fork ();
        printf ("pid = %d", getpid ());
        exit (0);
    }
    printf ("pid = %d", getpid ());
    return 0;
}

b) int main()
{
    fork ();
    if (fork () == 0)
    {
        fork ();
        printf ("pid = %d", getpid ());
    }
    printf ("pid = %d", getpid ());
    return 0;
}
```

Dla podpunktu a): 6 procesów, 6 linii pid = ...,
dla podpunktu b): 6 procesów, 10 linii pid =

19. Oblicz średni czas oczekiwania procesów na zasoby dla następującej kolejności procesów: P1 trwa 10 [ms], P2 trwa 8 [ms], P3 trwa 2 [ms] wg algorytmów:

1) FCFS,

2) SJF.

1) Dla algorytmu FCFS (First Come, First Served):

Kolejność procesów: P1 → P2 → P3 (niezmieniona)

Proces P1 zostanie obsłużony jako pierwszy, wykorzystując 10ms, proces P2 jako drugi, czekając 10ms i wykorzystując 8ms, proces P3 jako trzeci, czekając 18ms i wykorzystując 2ms.

Mamy więc:

$$0 + 10 + 18 = 28$$

$$\frac{28}{3} = 9,33 \text{ ms}$$

Średni czas oczekiwania procesów na zasoby wynosi 9,33ms.

2) Dla algorytmu SJF (Shortest Job First):

Kolejność procesów: P3 → P2 → P1 (zmieniona)

Proces P3 zostanie obsłużony jako pierwszy, wykorzystując 2ms, proces P2 jako drugi, czekając 2ms i wykorzystując 8ms, proces P1 jako trzeci, czekając 10ms i wykorzystując 10ms.

Mamy więc:

$$0 + 2 + 10 = 12$$

$$\frac{12}{3} = 4 \text{ ms}$$

Średni czas oczekiwania procesów na zasoby wynosi 4ms.

20. Omów algorytmy planowania dostępu do dysku.

SCAN (algorytm windy) – zamówienia są realizowane w kolejności mijania ścieżek przez głowicę wykonującą ruch wahadłowy od bieżącego miejsca do skrajnej ścieżki i z powrotem do ścieżki przeciwległej.

C-SCAN (Circular SCAN) – odmiana algorytmu SCAN, w której po dojściu do skrajnej ścieżki głowica wraca szybko do ścieżki przeciwległej, bez realizowania zamówień (uzasadnienie: po przeciwległej stronie nazbierało się zapewne więcej zamówień).

LOOK – ulepszenie algorytmu SCAN polegające na tym, że głowica nie podąża do skrajnych ścieżek, zawracając po obsłużeniu zamówienia z najdalszej (w danym przejściu) ścieżki.

Strategia C-LOOK stanowi modyfikację strategii LOOK. Podobnie, jak w przypadku C-SCAN, odwołania do dysku są realizowane tylko gdy głowice przesuwają się w jedną stronę po dysku. Po zrealizowaniu skrajnego odwołania głowice zawracają i przesuwają się do skrajnego odwołania po drugiej stronie dysku. Strategia ta jest trochę wolniejsza niż strategia look, ale za to średni czas oczekiwania na realizację odwołania jest taki sam dla wszystkich cylindrów.

21. Omów pierścieniowy algorytm wyboru koordynatora.

Algorytm pierścieniowy ma zastosowanie w systemach zorganizowanych w postaci pierścienia (logicznego lub fizycznego).

Zakłada się, że łączy się jednokierunkowe i że procesy wysyłają komunikaty zgodnie z ruchem wskazówek zegara. Procesy mogą ulegać awarii, każdy proces może się komunikować bezpośrednio z każdym innym w grupie. Główną strukturą danych jest lista aktywna zawierająca numery priorytetów wszystkich procesów aktywnych w systemie w chwili zakończenia algorytmu. Każdy proces utrzymuje własną listę aktywną.

Jeśli proces P_i wykrywa awarię koordynatora, to wysyła komunikat ELECTION ze swoim numerem do sąsiada. Jeśli sąsiad nie działa, to wysyła komunikat do następnego procesu wzdłuż pierścienia.

Na każdym kroku nadawca dodaje swój własny numer do listy w komunikacie.

Kiedy komunikat dociera do procesu, który go wysłał (proces rozpoznaje swój własny numer w komunikacie), to zostaje wybrany zwycięzca (proces z najwyższym numerem), a odbiorca komunikatu wysyła do sąsiada komunikat COORDINATOR w celu poinformowania wszystkich kto wygrał wybory oraz jakie procesy wchodzi w skład pierścienia i w jakiej kolejności. Kiedy ten komunikat okrąży pierścień, zostaje usunięty przez nadawcę.

22. Na czym polega buforowanie w odniesieniu do wczesnych systemów operacyjnych i w odniesieniu do podsystemu wejścia/wyjścia?

Buforowanie to metoda posiadająca dwa konteksty i analogiczne do nich znaczenia:

- 1) Buforowanie w odniesieniu do wczesnych systemów operacyjnych
 - metoda przetwarzania pośredniego, polegająca na tym, że dane z wejścia najpierw przepisywane są do bufora, z niego do jednostki centralnej, zaś odpowiedź znów najpierw do bufora, a następnie do wyjścia,
 - nie eliminuje całkowicie przestojów CPU czy urządzeń wejścia/wyjścia,
 - wymaga przeznaczenia pamięci na systemowe bufory,
 - niweluje wahania w czasie przetwarzania danych.
- 2) Buforowanie w odniesieniu do systemu wejścia/wyjścia - działa analogicznie, przy tym:
 - jeśli występują różnice w szybkości urządzeń, niezbędne jest podwójne buforowanie - przy odbieraniu oraz wysyłaniu informacji,
 - kolejnym problemem, rozwiązywanym przez sterowniki są różne rozmiary przesyłanych jednostek danych,
 - oraz zapewnienie semantyki kopii - operacje są wtedy wykonywane na buforze w jądrze.

23. Omów algorytm z Berkeley (synchronizacji zegarów).

Algorytm z Berkeley (brak odbiornika WWV!)

Sekwencja:

- 1) serwer czasu odpytuje maszyny,
- 2) serwer czasu oblicza średni czas,
- 3) wszystkie maszyny wyrównują czasy.

24. Omów metodę przydziału miejsca na dysku, która Twoim zdaniem najlepiej nadaje się do implementowania dostępu swobodnego.

Przydział indeksowy polega na zebraniu numerów bloków tworzących plik w jednym bloku, tzw. bloku indeksowym. Plik na dysku jest reprezentowany przez numer bloku indeksowego.

Przy zastosowaniu takiej strategii, dostęp do dowolnego miejsca w pliku wymaga co najwyżej dwóch operacji dyskowych - oznacza to bardzo krótki czas dostępu, co sprzyja implementowaniu dostępu swobodnego. Jednak wielkość pliku jest ograniczona. Przykładowo, jeżeli bloki dyskowe mają po 4KB, a numer bloku zajmuje 32bity, to maksymalna wielkość pliku to $1024 \cdot 4KB = 4MB$.

25. Omów metodę przydziału miejsca na dysku, która twoim zdaniem najgorzej nadaje się do implementowania dostępu swobodnego.

Najgorzej nadaje się metoda przydziału miejsca na dysku, która cechuje się powolnym dostępem do danych, a więc przydział listowy.

26. Na czym polega stronicowanie wykorzystujące tablicę haszowania?

27. Omów rozproszony algorytm wzajemnego wyłączenia.

Synchronizacja za pomocą rozproszonego algorytmu wzajemnego wyłączenia polega na tym, że proces, który chce wejść do sekcji krytycznej, wysyła do pozostałych procesów komunikat-zamówienie.

Pozostałe procesy:

- a) odsyłają komunikat OK, jeśli nie chcą teraz wchodzić do sekcji krytycznej,
- b) milczą i ustawiają zamówienie w kolejce, jeśli znajdują się w tej chwili w sekcji krytycznej,
- c) proces, który czeka na wejście do sekcji krytycznej porównuje znaczniki czasowe i odsyła OK, jeśli otrzymane zamówienie było wcześniej od jego własnego, zaś gdy było później – ustawia zamówienie w kolejce. Po skompletowaniu komunikatów OK od wszystkich procesów dany proces wchodzi do sekcji krytycznej, a po wyjściu wysyła do wszystkich procesów ze swojej kolejki komunikat OK (i usuwa je z kolejki).

28. Wymień algorytmy zastępowania stron, które są wolne od anomalii Belady'ego.

LRU (Least Recently Used)

OPT (optimal page replacement algorithm)

29. Policz średni czas oczekiwania dla algorytmu szeregowania zadań SJF, dla procesów o czasach trwania faz wynoszących kolejno: 4, 3, 2, 7, 1 jednostek.

Kolejność procesów: $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 7$ (zmieniona)

Mamy więc:

$$0 + 1 + 3 + 6 + 10 = 20$$

$$\frac{20}{5} = 4\text{ms}$$

Średni czas oczekiwania procesów na zasoby wynosi 4ms.

30. Różnice między wieloprogramowaniem a wielozadaniowością.

Wieloprogramowość (multiprogramming) opisuje zdolność komputera do przechowywania w pamięci operacyjnej kodu i danych wielu programów i przełączania się między tymi programami. Najprostszym przykładem są programy rezydentne dla systemu operacyjnego MS-DOS, takie jak SideKick. Program rezydentny po uruchomieniu kończy natychmiast swoją pracę, korzystając jednak z funkcji systemu kończącej proces bez usuwania go z pamięci operacyjnej. Ponieważ wcześniej jeszcze rejestruje też procedurę obsługi przerwania (na przykład przerwania klawiatury), jest w stanie przejąć sterowanie nad komputerem w momencie pojawienia się wyzwalającego go zdarzenia.

Jednoczesne przechowywanie kilku programów w pamięci nie implikuje jednak możliwości jednoczesnego ich działania, czyli wielozadaniowość (ang. multitasking). Wielozadaniowość w skrócie polega na okresowym, naprzemiennym przerywaniu i wznowianiu długotrwałych operacji w celu stworzenia iluzji jednoczesnej ich realizacji. Przykładem wielozadaniowości (aczkolwiek stosunkowo prymitywnej) jest program rezydentny buforujący dane wysyłane na drukarkę, dzielący je na małe bloki i wysyłający te bloki drukarce w ramach przerwy zegarowej, zupełnie niezależnie od realizowanego programu głównego.

31. Wyjaśnij, na czym polegają metody optymalizacji wykorzystania pamięci:

1) wymiana ,

2) ładowanie dynamiczne .

Optymalizacja wykorzystania pamięci:

- 1) Ładowanie dynamiczne - żądany podprogram jest ładowany w momencie wywołania.
- 2) Konsolidacja dynamiczna - w specjalnym obrazie binarnym 'stub' znajduje się namiastka procedury wskazująca jak odnaleźć podprogram biblioteczny.
- 3) Nakładki - zawierają moduły konieczne w danym momencie, do wywołania danego podprogramu. Są one projektowane przez programistę.
- 4) Wymiana - to rotacyjny, priorytetowy algorytm planowania. Działa on na zasadzie 'roll in - roll out' (ekspedytor - dispatcher). Charakteryzuje go długi czas przełączania kontekstu - 200ms (kwant czasu procesora - 0.2 s).

32. Zakładając, że rozmiar bloku wynosi n napisz wzór na maksymalny rozmiar pliku w ext2.

$$\min\left(\left(\left(\frac{n}{4}\right)^3 + \left(\frac{n}{4}\right)^2 + \frac{n}{4} + 12\right) * n, 2^{41}\right)$$

33. Wymień argumenty przemawiające za dużym rozmiarem strony.

Zmniejszanie rozmiaru strony zmniejsza fragmentację wewnętrzną, ale zwiększa zużycie pamięci na przechowywanie tablic stron.

Zwiększanie rozmiaru strony zmniejsza zużycie pamięci na tablice stron, ale powoduje większą fragmentację wewnętrzną pamięci.

34. Wymień zadania planisty średnioterminowego.

Planista średnioterminowy występuje w niektórych systemach z podziałem czasu. Jego zadaniem jest, w koniecznych przypadkach, zmniejszanie stopnia wieloprogramowości poprzez wysyłanie części zadań chwilowo na dysk (swapping). Pomaga to w doborze lepszego zestawu procesów w danej chwili, lub dla zwolnienia obszaru pamięci.

35. Omów stronicowanie wykorzystujące odwróconą tablicę stron.

W systemach w pełni 64-bitowych stosuje się tzw. odwrotną tablicę stron. Odwrotna tablica stron zawiera tyle pozycji, ile ramek pamięci fizycznej jest faktycznie zainstalowanych w komputerze (lub wielokrotność tej liczby, np. czterokrotność). W każdej pozycji tablicy pamiętana jest para: numer strony i numer ramki, w której ona się znajduje. Dzięki temu, wielkość odwrotnej tablicy stron jest proporcjonalna do wielkości pamięci fizycznej faktycznie zainstalowanej w komputerze i stanowi jej niewielki promil. Aby przetłumaczyć numer strony na numer ramki, nie przeglądając całej tablicy, można wykorzystać fakt, że odwrotna tablica stron jest tablicą mieszającą (ang. hash-table) -- strukturą słownikową o stałym oczekiwanym czasie dostępu. Dodatkowo, zwykle obliczanie funkcji mieszającej jest zaimplementowane sprzętowo w MMU, a odwrotnej tablicy stron towarzyszy szybka podręczna tablica stron.

36. Wyjaśnij pojęcia: sterownik, port wejścia/wyjścia, kanał wejścia/wyjścia, moduł sterujący (device driver).

Sterownik - układ elektroniczny, który nadzoruje pracę portu, szyny lub urządzenia albo większej ich grupy.

Port wejścia/wyjścia - punkt łączący między urządzeniem i komputerem.

Kanał wejścia/wyjścia - kanał, przy użyciu którego odbywa się przepływ danych.

Moduł sterujący - specjalne oprogramowanie bezpośrednio zarządzające sterownikiem urządzenia i ukrywające jego szczególne właściwości przed podsystemem wejścia/wyjścia w jądrze.

37. Wymień mechanizmy synchronizacji procesów, omów algorytm piekarni.

- 1) Synchronizacja programowa
 - niepodzielne operacje (zamiana, testuj i ustal)
 - zakaz przerwań (środ. 1-procesorowe, wieloprocessorowe)
 - aktywne oczekiwanie (busy waiting)
 - wirująca blokada (spinlock)
- 2) Semafore
 - mechanizm pierwotny
 - nie wymaga aktywnego oczekiwania
 - nie jest strukturalny
 - możliwość błędu w systemie (blokada)
- 3) Strukturalne mechanizmy synchronizacji
- 4) Algorytm piekarni
 - synchronizuje 2 lub więcej procesów,
 - proces, który chce wejść do SK bierze bilet (numerowany wg kolejności),
 - proces czeka, aż jego bilet będzie miał najniższą wartość.

38. Wymień wady rozproszonego algorytmu wzajemnego wyłączenia.

- uszkodzenie jednego procesu blokuje wszystkie próby wejścia do sekcji krytycznej (nie uda się skompletować komunikatów OK)
- duży ruch w sieci – każdy proces jest zaangażowany we wszystkie decyzje o wejściu do sekcji krytycznej
- algorytm jest wolniejszy, kosztowniejszy i mniej odporny od algorytmu scentralizowanego (ale jest możliwy, daje wzajemne wyłączanie bez blokad i zagłodzenia)

39. Zarządzanie wolną przestrzenią dyskową w systemie Linux.

W systemie operacyjnym Linux występuje mapa bitowa wolnych bloków w grupie. W przypadku gdy tworzony jest nowy plik – robi się to w punkcie początku grupy bloków, gdy chcemy plik rozszerzyć - od bloku przydzielonego ostatnio danemu plikowi. Metoda ta pozwala przydzielać wolne bloki w taki sposób, aby kolejne bloki pliku były (przeważnie) zgrupowane jako kolejne bloki na dysku, co znacznie przyspiesza dostęp do pliku.

40. Podaj jedną najważniejszą różnicę pomiędzy buforowaniem a spoolingiem.

Buforowanie - metoda jednoczesnego wykonywania obliczeń i wejścia-wyjścia dla jednego zadania

Spooling - metoda jednoczesnego wykonywania wejścia/wyjścia jednego zadania i obliczeń dla innego zadania

41. Wyjaśnij na czym polega spooling w odniesieniu do wczesnych systemów operacyjnych i w odniesieniu do podsystemu wejścia/wyjścia.

Spooling to metoda posiadająca dwa konteksty i analogiczne do nich znaczenia:

- 1) Spooling w odniesieniu do wczesnych systemów operacyjnych - spooling (simultaneous peripheral operation on-line) polega na przenoszeniu danych do szybszej pamięci zewnętrznej.

wejście - dysk - jednostka centralna - dysk - wyjście

Metoda ta pozwala na jednoczesne wykonywania wejścia/wyjścia jednego zadania i obliczeń dla innego. Spooling jest możliwy dzięki upowszechnieniu się systemów dyskowych - podczas wykonywania jednego zadania system operacyjny:

- czyta następne zadanie z czytnika kart na dysk (ustawia się kolejka zadań),
- drukuje umieszczone na dysku wyniki poprzedniego zadania.

Istnieje 'pula zadań' - powstaje możliwość wyboru kolejnego zadania do wykonania.

- 2) Spooling w odniesieniu do systemu wejścia/wyjścia - działa analogicznie, należy jednak podkreślić, że dla każdej aplikacji istnieje osobny bufor. Chroni to system przed niedopuszczalnym przeplotem danych w strumieniu (np. komunikacja z drukarką).

42. Omów algorytm Cristiana (synchronizacji zegarów).

Algorytm Cristiana jest przeznaczony głównie dla środowisk rozproszonych, w których jeden z węzłów jest serwerem czasu (time server) (np. posiada odbiornik WWV).

Zakłada się, że każda maszyna co pewien określony czas wysyła do serwera czasu zapytanie o podanie aktualnego czasu. Po otrzymaniu tego zapytania, serwer odpowiada najszybciej jak tylko może i przesyła aktualny czas UTC. Nadawca z kolei, po otrzymaniu informacji o czasie od serwera, zanim ustawi wartość swojego zegara, musi uwzględnić parę kwestii.

Mianowicie, zwykłe przepisanie czasu nadesłanego z serwera mogłoby spowodować, że czas płynie wstecz. Może się tak zdarzyć jeżeli zegar nadawcy wymierza czas zbyt szybko. Dodatkowo istnieje pewien koszt w postaci czasu komunikacji, który powoduje, że wartość czasu UTC wysłana przez serwer jest nieaktualna po nadejściu do nadawcy zapytania. Aby rozwiązać ten problem nadawca może np. zapamiętać przedział czasowy zawarty pomiędzy momentem T_0 , w którym wysłano zapytanie do serwera i momentem T_1 , kiedy przyszła odpowiedź z serwera.

43. Omów algorytm Lamporta (synchronizacji zegarów).

Algorytm lamporta służy do synchronizacji procesów w systemach rozproszonych. Pozwala on rozwiązać problem wzajemnego wykluczania dla n procesów.

Działa w następujący sposób: komunikat wysyłany przez każdy proces zawiera czas swojego nadania, przy czym czas przesłania komunikatu jest większy bądź równy jeden. Jeżeli czas przybycia komunikatu jest mniejszy lub równy czasowi nadania, to odbiorca przesuwa czas zegara na wartość równą czasowi nadania, powiększoną o 1 (jeden). gdzie dodatkowym warunkiem jest że żadne dwa zdarzenia nie powinny mieć tego samego czasu. Algorytm zapewnia całkowite uporządkowanie wszystkich zadań w systemie rozproszonym.

44. Gdzie definiuje się aliasy i dlaczego właśnie tam?

Zdefiniowane aliasy obowiązują w sesji powłoki. Jeżeli są wpisane do plików konfiguracyjnych powłoki (np. `~/.cshrc`, `/etc/csh.cshrc`, `~/.bashrc`, `/etc/bashrc`), będą dostępne przy każdym uruchomieniu powłoki.

45. Na czym polega wymiana (swapping)?

Wymiana (swapping) polega na wirtualnym powiększeniu dostępnej pamięci operacyjnej poprzez stworzenie pamięci wirtualnej. Pamięć używana przez procesy jest dzielona na strony, które mogą być przechowywane w pamięci operacyjnej lub też w przestrzeni wymiany na dysku twardym. Gdy zaczyna brakować miejsca w pamięci operacyjnej, niektóre nieużywane strony (według odpowiednich algorytmów) są odsyłane na dysk – dzięki temu zwalnia się nieco miejsca w pamięci RAM. Gdy strony te będą potrzebne innemu procesowi, zostaną z powrotem sprowadzone do pamięci operacyjnej.

46. Na czym polega zasada wiedzy koniecznej?

Zasada wiedzy koniecznej to zalecenie projektowe, ważne przy budowie systemów operacyjnych, w myśl którego procesy w każdej chwili działania powinny dysponować tylko niezbędnymi prawami dostępu do zasobów. Na przykład proces, którego zadaniem jest czytanie pliku, nie powinien mieć prawa zapisywania tego pliku. Przestrzeganie zasady wiedzy koniecznej zmniejsza skutki awarii systemu.

47. Co to jest „koń trojański”?

To oprogramowanie, które podszywając się pod przydatne lub ciekawe dla użytkownika aplikacje dodatkowo implementuje niepożądaną, ukrytą przed użytkownikiem funkcjonalność np. spyware, bomby logiczne itp.

Do głównych zagrożeń należy: szpiegowanie i wykradanie poufnych danych użytkownika, rozsyłanie spamu, utrudnianie pracy programom antywirusowym, usuwanie danych.

48. Wymień linie obrony w systemach uniksowych.

- 1) Ochrona fizyczna (dostęp do komputera)
- 2) Hasła
- 3) Prawa dostępu do plików
- 4) Szyfrowanie plików
- 5) Kopie zapasowe
- 6) Komercyjne systemy dostarczają również narzędzi zabezpieczających system.

49. Co to jest 'bomba logiczna'?

Jest to fragment kodu programu komputerowego wykonujący pewne instrukcje po spełnieniu określonych warunków. Głównym zagrożeniem związanym z bombami logicznymi jest ich nieaktywność, a co za tym idzie - trudna wykrywalność do czasu zaktywizowania.

Warunkami uruchomienia bomby logicznej mogą być:

- 1) określona data/godzina/dzień tygodnia,
- 2) uruchomienie programu,
- 3) połączenie z internetem.

50. Jakie są zagrożenia programowe?

- 1) używanie programów pisanych przez innych użytkowników,
- 2) konie trojańskie,
- 3) boczne wejście – czyli pozostawienie luki w oprogramowaniu przez projektanta,
- 4) obchodzenie procedur bezpieczeństwa dla pewnego użytkownika,
- 5) udostępnianie sobie haseł dostępu.

51. Jakie są zagrożenia systemowe?

- 1) bakterie – używają mechanizmu rozmnażania aby sparaliżować działanie systemu,
- 2) robaki – przechodzą z systemu na system; rozprzestrzenia się we wszystkich sieciach podłączonych do zarażonego komputera poprzez wykorzystanie luk w systemie operacyjnym lub naiwności użytkownika. Może mieć wbudowane procedury dodatkowe, takie jak niszczenie plików, wysyłanie poczty (z reguły spam) lub pełnienie roli backdoora lub konia trojańskiego,
- 3) wirusy – najczęściej proste programy komputerów, które w sposób celowy powielają się bez zgody użytkownika. Wirus komputerowy w przeciwieństwie do robaka do swojej działalności wymaga nosiciela w postaci programu komputerowego, poczty elektronicznej, itp. Wirusy wykorzystują słabość zabezpieczeń systemów komputerowych lub właściwości systemów oraz niedoświadczenie i beztroskę użytkowników.

52. Z czego wynika ograniczenie na liczbę wpisów w katalogu głównym w systemie FAT?

53. Na czym polega szamotanie i jakie są jego przyczyny?

Szamotanie to duża aktywność wystąpień stronicowania. Mówi się, że proces szamoce się, jeśli więcej czasu spędza na stronicowaniu niż na wykonaniu.

Przyczyną występowania szamotania może być zła synchronizacja programów korzystających ze stronicowania. Pierwszy program wymusza stronicowanie drugiego, a drugi powoduje stronicowanie pierwszego, co prowadzi do znacznego obniżenia wydajności systemu.

54. Jak można zapobiegać szamotaniu?

Jedną z metod zapobiegania szamotaniu jest wyznaczanie tzw. zbiorów roboczych procesów. Zbiór roboczy, to zbiór stron, do których nastąpiło odwołanie w ciągu ostatnich I instrukcji (dla pewnej ustalonej stałej I , np. $I = 100\ 000$). Zbiór roboczy ma przybliżać strefę, w której znajduje się program.

Należy dobrze dobrać parametr I , ponieważ jego zbyt mała wartość uniemożliwi uchwycenie całej strefy, a zbyt duża spowoduje, że łączy się rozmiary kilku stref. Znajac wielkości zbiorów roboczych, możemy przydzielić procesom pamięć proporcjonalnie do wielkości ich zbiorów roboczych.

Jeśli suma rozmiarów zbiorów roboczych wszystkich procesów jest większa niż rozmiar dostępnej pamięci, prawdopodobnie mamy do czynienia z szamotaniem. Należy wówczas wstrzymać jeden z procesów, aby nie pogarszać sytuacji.

55. Co to jest wirująca blokada (spinlock)?

Spinlock, inaczej wirując blokada lub czekanie aktywne, to oczekiwanie przez proces na wystąpienie pewnego zdarzenia, podczas którego proces nie zwalnia procesora, lecz do czasu przerwania wykonuje „martwą” pętlę. Jest to rozsądne rozwiązanie w systemach wieloprocesorowych. W architekturach jednoprosesorowych czekanie aktywne jest nazbyt kosztowne i na ogół stosuje się zamiast niego usypianie procesu.

56. Co to jest proces zombie? Napisz fragment kodu tworzący taki proces.

Proces zombie to wpis w tablicy procesów opisujący program, którego wykonanie w systemie operacyjnym zostało zakończone, ale którego zamknięcie nie zostało jeszcze obsłużone przez proces rodzica.

```
#include <stdio.h>
main ()
{
    if (fork () == 0)
        exit (0);
    sleep (30);
    wait (NULL);
}
```

57. Na czym polega wyższość RAID5 nad RAID3 lub odwrotnie?

RAID (Redundant Array of Independent Disks)

- 1) RAID0 – striping (paskowanie; przemiennie ułożone paski z kolejnych urządzeń fizycznych tworzą jedno urządzenie logiczne – większa szybkość)
- 2) RAID1 – mirroring (jeden dysk dokładną kopią drugiego – polepszenie bezpieczeństwa)
- 3) RAID3 – paskowanie z danymi o parzystości (XOR), dane o parzystości na jednym dysku
- 4) RAID5 – paskowanie z danymi o parzystości (XOR), dane o parzystości rozproszone pomiędzy wszystkie dyski (co najmniej 3) – możliwość rekonstrukcji danych na jednym dysku w oparciu o dane na drugim dysku i dane parzystości

Idea RAID3 i RAID5 jest podobna, ale w RAID5 dane o parzystości są rozproszone pomiędzy wszystkie dyski wchodzące w skład macierzy, podczas gdy w RAID3 są one przechowywane na jednym wybranym dysku. Ten dysk jest wówczas znacznie bardziej intensywnie używany od pozostałych (zmiana jakichkolwiek danych wymaga uaktualnienia danych o parzystości, a więc zapis na tym wybranym dysku). Mankament ten rozwiązano w RAID 5 – dzięki rozbiciu danych o parzystości pomiędzy kilka dysków udało się zrównoważyć intensywność używania wszystkich dysków w macierzy).

58. Co to jest MBR (Master Boot Record)?

Master Boot Record (Master Boot Record) – umowna struktura zapisana w pierwszym sektorze dysku twardego (i dyskietki). Zawiera on program rozruchowy oraz główną tablicę partycji.

MBR ma 512 bajtów długości. Znajduje się na pierwszej ścieżce, w pierwszym cylindrze, w pierwszym sektorze dysku.

59. Wyjaśnij pojęcie DMA (Direct Memory Access).

DMA (Direct Memory Access) - technika, w której inne układy (np. kontroler dysku twardego, karta dźwiękowa, itd.) mogą korzystać z pamięci operacyjnej RAM lub (czasami) portów wejścia/wyjścia, pomijając przy tym procesor główny - CPU.