Przedmiot WSO 1
Pytania egzaminacyjne podstawowe

1. Co to jest system operacyjny?
System operacyjny jest programem, który działa jako pośrednik między użytkownikiem komputera a sprzętem.
Program, który nadzoruje i koordynuje dostęp programów do zasobów (CPU, HDD, RAM, inne)

2. Co oznacza wielodostępność, a co wielozadaniowość?
* wielozadaniowość - można wykonywać wiele zadań (procesów) jednocześnie i cały czas sprawować nad nimi kontrolę
* wielozadaniowość z wywłaszczaniem - błędy powstałe podczas wykonywania jednej aplikacji nie wpływają na
działanie innych i nie powodują destabilizacji systemu
* wielodostępność -możliwość pracy kilku użytkowników w jednym czasie

3. Jakie funkcje spełnia wielodostępny system operacyjny i jakie podsystemy wchodzą w jego skład?
Pośredniczy pomiędzy sprzętem a użytkownikiem.
Podsystemy OS'a:
* programy obsługi sprzętu (sterowanie sprzętem)
* jądro (kernel) - zarządzanie procesami, pamięcią, systemem plików
* interfejs programów użytkowych - odwołania do systemu (API - application program interface)
* powłoka (shell)
* programy systemowe (polecenia)
* programy użytkowe

4. Co jest przedmiotem standaryzacji w wielodostępnych systemach operacyjnych?
Cele standaryzacji:
* przenaszalność - łatwe przenoszenie OS'a z jednego systemu operacynego na inny
* współpraac aplikacji - wymiana informacji między aplikacjami pracuącymi na różnych odmianach OS'a
* skalowalność - elastyczna rozbuowa systemu i powiększanie jego możliwości wraz z rozbudową aplikacji
Istota standardu: określenie interfejsu, a nie impementacji.

5. Co to jest proces?
Proces to twór abstrakcyjny, który składa się z wykonywanego programu oraz bieżących danych o jego stanie
i zasobach za pomocą których system operacyjny steruje jego wykonywaniem.
Pamięć procesu można podzielić na trzy segmenty :
- segment instrukcji, który zawiera kod wykonywalny programu oraz stałe. W Unixach segment ten jest
tylko do odczytu i może byc wspólny dla wielu procesów (np. Wówczas, gdy uruchomimy ten sam program kilkakrotnie)
- segment danych - w którym znajdują się dane zainicjowane i niezainicjowane(zmienne statyczne i globnalne),
- segment stosu - w którym przechowywane są zmienne automatyczne, rejestrowe i dane związane z wywołaniem funkcji.

6. Co to jest blok kontrolny procesu i do czego służy?
Każdy proces jest reprezentowany w systemie operacyjnym przez swój blok kontrolny procesu. Blok kontrolny
procesu jest blokiem danych lub rekordem zawierającym wiele cząstkowych informacji opisujących dany
proces takich jak:
* stan procesu
* licznik rozkazów
* rejestry procesora
* informacje o planowaniu przydziału procesora
* informacje o zarządzaniu pamięcią

* informacje do rozliczeń
* informacje o stanie wejścia-wyjścia

Blok kontrolny procesu służy jako magazyn przechowujący każdą informację, która może się zmieniać w zależności od procesu.

7. Co oznacza współbieżne wykonywanie procesów?

Jednoczesne wykonywanie wielu procesów.
Istotne zagadnienia współbieżności powstają wtedy, gdy procesy są od siebie zależne, tzn. gdy w różnych procesach występują zdarzenia takie, że warunkiem zajścia jednego z nich jest zajście drugiego i odwrotnie.

8. Jak powstaje nowy proces?

"Ale jak właściwie proces powstaje? Prześledźmy to (w uproszczeniu) na przykładzie komendy ls.

1. Komendę tę wpisujemy w shell'u. Jest on więc tzw. procesem macierzystym ponieważ na jego bazie powstaje nowy proces. A chyba nikt nie wątpi że sam shell jest także procesem? Przecież jest to uruchomiony program.
2. Po wprowadzeniu komendy ls shell tworzy w pamięci operacyjnej kopię samego siebie wraz z tzw. środowiskiem. Do środowiska należą takie rzeczy jak zmienne oraz informacje o prawach dostępu danego użytkownika.
3. Następnie shell zastępuje swoją kopię przez program ls. Powstaje w ten sposób nowy proces ls. Jest to tzw. proces potomny shell'a.
4. Proces ten znajduje się w środowisku w jakim umieścił go shell. Dziedziczy więc np. wszystkie prawa (i ograniczenia) jakie posiada użytkownik uruchamiający komendę. Każdy proces ma więc także swojego właściciela tak jak to jest w przypadku plików.
5. W tej chwili proces ls wykonuje się. Robi to co do czego został stworzony a więc przegląda katalog i wyświetla na ekranie jego zawartość. W tym czasie shell czeka na jego zakończenie. Czasem mówimy, że "śpi". To właśnie dlatego nie mamy w tym czasie dostępu do linii poleceń i nie możemy nic napisać.
6. Po zakończeniu swojego działania kernel sygnalizuje shell'owi że proces ls się już skończył a następnie usuwa go z pamięci operacyjnej. Shell na ten sygnał się budzi i prezentuje nam nowy znak dolara.

Możemy wpisać następną komendę."

10. Jak działa funkcja systemowa exec?

execl, execlp, execle, execv, execvp - wykonanie pliku
Rodzina funkcji exec() zastępuje w pamięci obraz aktualnego procesu obrazem nowego procesu.
Wywołuje program wskazany przez filename. filename musi być albo programem wykonywalnym, albo skryptem powłoki, rozpoczynającym się od linii "#! interpreter [arg]". W tym drugim przypadku interpreter musi być prawidłowym programem, a nie skryptem, który będzie uruchomiony jako interpreter [arg] filename.

11. Jak działa funkcja systemowa fork?

fork - utwórz proces potomny
fork tworzy proces potomny, który różni się od procesu rodzicielskiego jedynie swoim PID-em i PPID-em, i faktem, że użycie zasobów jest ustawione na 0. Locki plików i oczekujące sygnały nie są dziedziczone.

12. Co to są funkcje systemowe?

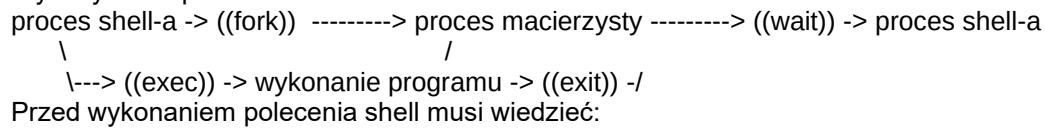
Funkcje systemowe: fork(2), execl(3), execlp(3), execv(3), execvp(3), execle(3), getpid(2), getppid(2), getgid(2), system(3) - obsługują procesy.

13. W jakich stanach może być proces?
* nowy - proces jest tworzony
* wykonywany - wykonywane są instrukcje:
- w trybie użytkownika
- w trybie jądra
* gotowy do wykonania - proces czeka na przydział procesora przez program szeregujący, w tym stanie jest zwykle wiele procesów
* czekający (uśpiony) - proces czeka na wystąpienie zdarzenia niezbędnego do jego dalszego wykonania, np. na zakończenie operacji we/wy
* zakończony - proces zakończył wykonywanie

14. Co to jest "shell"?
Program pośredniczący między jądrem, systemem plików i programami usługowymi.
Działanie:
1. Użytkownik: wprowadza polecenie
2. Shell:
* czyta wiersz polecenia
* przetwarza wiersz polecenia
* tworzy proces
3. Unix: wykonue proces

15. Podać przykłady programów shell i ich właściwości?
* Shell Korna, Bourna, C Shell.
Właściwości:
* Przekazywanie starowania do programu wybranego poleceniem użytkownika
* Wykonywanie wbudowanych poleceń
* Dostarczanie języka do pisania skryptów
* Ustawianie środowiska pracy
* Przywoływanie i edycja uprzednio wydanych poleceń
* Przeadresowywanie wejścia-wyjścia poleceń
* Generowanie nazw plików
* Umożliwienie łączenia poleceń w potok
* Umożliwienie przetwarzania w drugim planie

16. W jaki sposób program shell interpretuje polecenie?
Shell dopuszcza 3 typy poleceń:
* plik wykonywalny, który zawiera kod wynikowy skompilowanego programu
* plik wykonywalny, który zawiera ciąg poleceń la shell-a (skrypt)
* wewnętrzne polecenia shell-a
Schemat wykonywania poleceń:



Przed wykonaniem polecenia shell musi wiedzieć:
* kto wykonuje polecenie (czy ma prawo do wykonania)
* gdzie umieszczone jest polecenie do wykonania

17. Na czym polega wykonywanie polecenia w tle (w drugim planie)?
Interpretator rozpoznaje w wierszu polecenia znak &. Ustawia odpowiednią zmienną lokalną.
Pod koniec pętli sprawdza jej wartość i jeśli jest ustawiona to nie wykonuje funkcji wait, ale przechodzi do początku pętli i wczytuje następne polecenie.

18. Co to jest planowanie (szeregowanie) procesów?
* System z podziałem czasu - jądro przydziela procesor gotowemu procesowi na jeden kwant czasu, zgodnie z algorytmem rotacyjnym.
* Po upływie tego czasu wywłaszcza proces i przydziela procesor procesowi następnemu w kolejce. Wybiera proces załadowany do pamięci o najwyższym priorytecie.
* Jądro okresowo przelicza i modyfikuje priorytety wszystkich procesów.

* Wywłaszczony proces jest umieszczany w ednej z kolejek priorytetowych i gdy przyjdzie na niego
kolej, jest wznawiany od miejsca, w któ?ym nastąpiło zawieszenie wykonywania.

19. Podać przykłady algorytmów szeregowania procesów i wyjaśnić ich działanie?
Przykłady:
* FCFS (First Come, First Served) - pierwszy nadszedł pierwszy obsłużony. Realizowany za pomocą kolejki FIFO. Wady: efekt konwoju, kłopotliwy w systemach z podziałem czasu. Jest algorytmem niewywłaszczającym.
* SJF (Shortest Job First) - najpierw najkrótsze zadanie. Procesor przydzielany jest procesowi o najkrótszej następnej fazie procesora. Teoretycznie daje minimalny średni czas oczekiwania. Wymaga jednak dokładnego oszacowania czasu przyszłej fazy procesora dla każdego procesu. Szacowanie to wykonywane jest zwykle na podstawie pomiarów czasu faz poprzednic. Może być wywłaszczającym lub nie.
* Planowanie priorytetowe - każdemu procesowi przydzielany jest pewien priorytet. Wybierany jest proces o najwyższym. Algorytm SFJ jest szczególnym przypadkiem. Może być wywłaszczającym lub nie.
* Planowanie rotacyjne - zaprojektowany dla systemów z podziałem czasu. Ustalony jest kwant czasu (10-100ms). Kolejka procesów ma charakter cykliczny. Kolenym procesom przydzielany jest co najwyżej kwant czasu.

Działanie:
1. Wybierz proces o najwyższym priorytecie spośród procesów gotowych do wykonania i załadowanych do pamięci operacyjnej.
2. Jeśli jest kilka o równym priorytecie wybierz proces najdłużej oczekujący w stanie gotowym do wykonania.
3. Jeśli nie ma procesów gotowych do wykonania, czekaj do następnego przerwania (najpóźniej do kolejnego taktu zegara). Powóć do kroku 1.

20. Jak rozwiązuje się zagadnienie planowania procesów w systemach typu UNIX?
Planowanie polega na określaniu w akiej kolejności procesy uzyskują dostęp do zasobów komputera: CPU, RAM. Celem planowania jest zapewnienie jak najlepszego wykorzystania procesora. Procesy starając się o dostęp do procesora czy konkretnego urządzenia czekają w kolejkach: zadań, procesów gotowych, do urządzeń we/wy.

Kryteria planowania:
1. Wykorzystanie procesora, w % czasu.
2. Przepustowość - liczba procesów kończących w jednostce czasu.
3. Czas cyklu przetwarzania - średni czas przetwarzania procesu od chwili utowrzenia procesu.
4. Czas ozcekiwania - średni czas oczekiwania w kolejkach.
5. Czas odpowiedzi - w systemach interakcynych czas między zgłoszeniem zamówienia przez użytkownika a pierwszą odpowiedzią.

21. Kiedy, po co i jak wykonywany jest proces ładowania systemu?
Cel: umieszczenie OS'a w pamięci operacyjnej i rozpoczęcie jego wykonywania.

Etapy:

1. Inicjalizacja i testowanie sprzętu.
2. Wczytanie i umieszczenie w oamięci bloku systemowego (blok 0) z dysku.
3. Program zawarty w bloku systemowym ładuje jądro Os'a z pliku systemowego do pamięci. Przekazuje
sterowanie dopierwszej instrukcji jądra. Program jądra zaczyna się wykonywać.
4. Wykrywanie i konfiguracja urządzeń, odnalezienie głównego katalogu plików.
5. Przygotowanie środowiska procesu 0. Wykonywanie programu systemu jako procesu 0 wykonywanego
w trybie jądra. Utworzenie procesów jądra, np. procesów zarządzania pamięcią.
6. Rozwidlenie procesu 0. Utworzony proces 1 tworzy konkekst poziomu użytkownika i przechodzi
do trybu użytkownika.
7. Proces 1 wywołuje funkcję systemową exec wykonując program /sbin/init.
8. Proces init wczytuje plik /etc/inittab i rozmnaża procesy.
9. Inicjalizacja wewnętrznych struktur danych jądra, tworzenie np. listy wolnych buforów, i-węzłów,
kolejek; inicjalizacja struktur segmentów i tablic stron pamięci.
10. Sprawdzenie głównego i pozostałych systemów plików (ew. uruchomienie fsck).
11. Wywoływane są procesy getty monitorujące konsolę i terminale systemu komputerowego, zgodnie
z deklaracją wpliku inittab, a proces init wykonuje funkcję systemową wait monitorując zakończenie
procesów potomnych. Proces init tworzy również procesy demony.

22. Jaki proces ma PID=1, jakie zadania wykonuje?
Jest to init - steruje procesem inicjującym.
Init jest przodkiem wszystkich procesów. Jego głównym zadaniem jest stworzenie procesów w oparciu o skrypt zapisany w pliku /etc/inittab. Kontroluje też samodzielne procesy, których istnienie jest konieczne w danym systemie. Odpowiada też za pracę systemu w odpowiednim trybie (RUNLEVEL).

23. Co to są pliki i jakie typy plików występują w systemie UNIX?
Jednostka logiczna przechowywanej informacji, niezależna od właśiwości fisycznych urządzeń pamięciowych.
Zwykle w plikach przechowywane są programy i dane.
Podstawowe typy plików:
* pliki zwykłe
* pliki specjalne
* katalogi
* dowiązania symboliczne
* potoki nazwane FIFO (ang.named pipe)
* gniazda (ang. UNIX-domain sockets)

24. Co to jest i-węzeł?
Przyporządkowany plikowi rekord przchowujący większość informacji o pliku. Tworzone są w chwili tworzenia
systemu plików. Ich liczba zależy od rozmiaru oraz założonego średniego rozmaru pliku.

25. Jakie informacje są przechowywane w i-węźle?
Zawartość i-węzła:
* typ pliku
* prawa dostępu do pliku
* liczba dowiązań
* id właściciela
* id grupy
* rozmiar w bajtach
* czas ostatniej modyfikacji
* czas ostatniego dostępu
* czas ostatniej zmiany informacji w i-węźle

* 12 wskaźników zawierających adresy bloków z danymi pliku (bloki bezpośrednio adresowane).
* wskaźnik zawierający aders bloku, w którym przechowywane są adresy bloków z danymi (adresowanie pośrednie
jednostopniowe)
* wskaźnik adresowania pośredniego dwustopniowego
* wskaźnik adresowania pośredniego trójstopniowego

26. Jakie informacje przechowywane są w pliku typu "katalog"?
Przechowywane są nazwy plików łącznie z numerami odpowiadającym tym plikom i-węzłów.

27. Co to jest katalog z punktu widzenia użytkownika, a co z punktu widzenia budowy systemu plików?
Z punktu widzenia użytkownika jest to zbiór plików i innych katalogów.
Z punktu widzenia systemu plików to plik zawierający tablicę: nazwa_pliku<->i-węzeł.

28. Jaka jest różnica między adresowaniem bezpośrednim a adresowaniem pośrednim?
W adr. bezpośrednim 12 wskaźników zawiera bezpośrednie odnośniki do bloków z danymi pliku. Samo byłoby
nieefektywne i o znacznie ograniczałoby rozmiary plików.

29. Wyjaśnić mechanizm rozmieszczania bloków i fragmentów pliku w blokach na dysku?
Reguły przydzielania bloków i fragmentów:
1. Jeśli rozmiar pliku est mniejszy niż rozmiar fragmentu, plikowi temu przydzielany jest pierwszy wolny fragment.
2. Jeśli rozmiar pliku jest większy niż rozmiar fragmentu, alemniejszy niż rozmiar bloku, plikowi temu przydzielane są kolejne fragmenty należące do tego samego bloku.
3. Jeśli rozmiar pliku jest większy niż rozmiar bloku, plikowi temu przydzielana jest odpowiednia liczba bloków, niekoniecznie znajdująca się obok siebie, o łącznym rozmiarze nieprzekraczaącym rozmiaru pliku. Pozostała część pliku umieszczana jest w położonych obok siebie fragmentach należących
do jednego bloku zgodnie z regułami 1 oraz 2.

30. Jak adresowane są bloki i fragmenty pliku na dysku?
?
31. Czym różni się przydział ciągły miejsca na dysku od przydziału listowego?
?
32. Co jest tablicą FAT?
?
33. Porównać przydział listowy miejsca na dysku z przydziałem indeksowym?
?

34. Co to jest i jak jest wykorzystywana pamięć operacyjna?
Zainstalowana pamięć operacyjna nazywana jest pamięcią fizyczną.
Można ją interpretować jako tablicę bajtów, w której każdy element ma przyporządkowany jednoznaczny adres.
Wykonywane programy wraz z danymi (procesy) przechowywane są przynamniej częściowo w pamięci. Część pamięci
fizycznej, w której umieszczane są procesy nazywana jest pamięcią dostępną dla procesów.

35. Wyjaśnić mechanizm wymiany (swapping) procesów?
?

36. Wyjaśnić mechanizm stronicowania na żądanie?
Wykorzystuje zasadę lokalności odwołań i do pamięci przesyłane są strony niezbędne w danej chwili lub te,
które niebawem mogą okazać się niezbędne. Strony mogą więc znadować się na dysku, w pamięci głównej lub
w obrzarze wymiany. Jeśli niezbędna jest strona znadująca się na dysku, generowany jest błąd strony

i odpowiednie przerwanie. Wykonywanie procesu jest wstrzymywane, po czym odnajdywana jest wolna ramka,
do której przepisywana est potrzebna stron. Zwykle wymaga to zwolnienia ramki przez stronę innego procesu (wymiana stron, ang. paging) - niezbędne więc są odpowiednie algorytmy zastępowania stron.

37. Wyjaśnić mechanizm segmentacji?
Segmenty to semantycznie określone fragmenty programu, np. program główny, podprogramy i biblioteki,
tablica symboli, dane, stos. do podstawowych zalet segmentacji należą: możliwość powiązania ochrony pamięci
z wybranymi segmentami oraz współdzielenie wybranych segmentów przez różne procesy. Na przykład ustawienie
bitu ochrony dla segmentów kodu (tylko do odczytu) lub współdzielenie kodu edytora (np. vi) przez procesy
edycji wielu jednocześnie pracujących użytkowników. W tym przypadku obrazy procesów użytkowników zawierają,
w uproszczeniu, segmenty danych i stosu oraz wskaźniki do właściwego miesca w segmencie programu.

38. Podać przykłady algorytmów wymiany stron?
* Algorytm FIFO
* Algorytm optymalny
  Zastąp stronę, która nadłużej nie będzie używana.
* Algorytm LRU (ang. Last Recently Used)
  Zastąp stronę która nie była używana od najdłuższego czasu.

39. Kiedy występuje błąd strony i jak jest obsługiwany?
Występuje w przypadku braku strony w zbiorze roboczym. Jest to zbiór stron procesu jednocześnie znadujących się w pamięci.

40. Na czym polega zarządzanie pamięcią wykonywane przez system operacyjny?
41. Jakie są zadania podsystemu wejście - wyjście?
42. Co to są pliki specjalne i do czego służą?
Zapewniają aplikacjom i użytkownikom współpracę z urządzeniami zewnętrznymi.

43. Co to jest tablica rozdzielcza urządzeń we-wy, jakie informacje zawiera i do czego służy?
44. Jak wykorzystywana jest pamięć podręczna do wydajniejszego korzystania z systemów dyskowych?

45. Jakie są typy plików specjalnych?
* Blokowe
* Znakowe

46. Jakie informacje są zapisane w plikach specjalnych i do czego służą?
Charakterystyka:
* położenie w strukturze katalogów (/dev/fd0)
* określone zasady nazywania
* liczba główna, np 17
* liczba pomocnicza, np. 0x014000

47. Jaką rolę pełnią podprogramy obsługi urządzeń?
Inaczej drajwery (drivers) - moduły jądra przeznaczone do sterowania pracą urządzeń zewnętrznych.

48. Co to jest pamięć współdzielona?

Opracowanie - D.Makowski@wsisiz.edu.pl. W razie błędów, niejasności proszę o komentarze.
<-powrót