

#### 1. Co to jest system operacyjny?

System operacyjny to program zarządzający pracą komputera. Program pośredniczący między komputerem a użytkownikami. Zadania użytkownika wykonywane są jako procesy obsługiwane przez system operacyjny. System przydziela procesom zasoby komputera: dostęp do jednostki centralnej, pamięci, dyski i inne.

#### 2. Co oznacza wielodostępność, a co wielozadaniowość?

Wielodostępność oznacza to, że w jednej chwili wielu użytkowników może korzystać z systemu. Wiele użytkowników może być jednocześnie zalogowanych. Wielozadaniowość oznacza to, że w jednej chwili użytkownik może uruchomić wiele programów.

#### 3. Jakiej funkcje spełnia wielodostępny system operacyjny i jakie podsystemy wchodzi w jego skład?

System składa się z podsystemów wykonujących zadania:

- zarządzanie procesami
- zarządzanie pamięcią
- zarządzanie plikami
- zarządzanie przestrzenią dyskową
- system ochrony
- praca sieciowa.

#### 4. Co jest przedmiotem standaryzacji w wielodostępnych systemach operacyjnych?

Celem standaryzacji jest budowa systemów „otwartych” charakteryzujących się przenośnością aplikacji, możliwością współpracy oprogramowania na różnych maszynach, skalowalność skalowalność ę, możliwość rozbudowy sprzętowej i aplikacji bez potrzeby zmian systemu.

#### 5. Co to jest proces?

Proces to wykonywany program, tzn. program, który został rozpoczęty i jeszcze nie zakończył się. Proces składa się z kodu programu i przydzielonemu mu obszarowi pamięci.

#### 6. Co to jest blok kontrolny procesu i do czego służy?

Jest unikatowy dla każdego procesu i umożliwia nadzoru nadzoru nadzoru nadzoru. Zawiera „wskaźnik, stan procesu, numer procesu, licznik rozkazów, rejestry, ograniczenia pamięci, wykaz otwartych plików.

#### 7. Co oznacza współbieżne wykonywanie procesów?

W systemie wielodostępnym, pracującym z podziałem czasu jest jednocześnie wiele procesów. Procesy działają współbieżnie, co oznacza konieczność podziału obliczeniowej procesora (przełączanie) między poszczególne procesy, ponieważ każdy chwil czasu wykonany jest tylko jeden proces.

#### 8. Jak powstaje nowy proces?

Nowy proces w systemie jest uruchamiany w następujący sposób. Po pierwsze, istniejący proces wykonuje czynność zwaną rozwidleniem (forking), polegającym na utworzeniu dokładnej kopii samego siebie. Utworzony w ten sposób proces, określony jako proces dziecko, posiada własny identyfikator PID, chociaż jego środowisko jest dokładnie takie samo jak procesu rodzica. W tym momencie, w wyniku wykonywania funkcji systemowej o nazwie `exec`, istniejący program zawiera w przestrzeni adresowej procesu dziecka ulega wymianie na treść tego programu, który ma być wykonywany w ramach danego procesu. Treść nowego programu (czy polecenia) kompletnie zastępuje tę odcieżdzoną od procesu rodzica. Bez zmian pozostają jednak: środowisko procesu, w tym ustawienia zmiennych, przypisanie priorytetu, standardowego wejścia, standardowego wyjścia oraz standardowego wyjścia błędów, a także priorytety procesu.

#### 9. Jak proces jest kończony?

Momencie, gdy proces kończy swoje działanie, proces rodzic zostaje o tym poinformowany przez odpowiedni sygnał. Tak więc w przypadku, gdy użytkownik kończy sesję, login shell wysyła do procesu int sygnał informujący go o tym, że właśnie kończy działanie. Int podejmuje wówczas działanie umożliwiające zalogowanie się na zwinionym terminalu innemu użytkownikowi systemu, polegające na rozwidleniu się i wykonaniu programu `getty`.

#### 10. Jak działa funkcja systemowa `exec`?

Funkcja `exec` (executing command) służy do przekształcania nowego procesu w proces wykonywujący polecenie.

#### 11. Jak działa funkcja systemowa `fork`?

Funkcja `fork` (forking a process) tworzy nowy proces (proces potomny), który ma wykonać polecenie. Jest to kopia bieżącego procesu.

#### 13. W jakich stanach może być proces?

Stany procesów:

- nowy – proces jest tworzony
- wykonywany – w trybie użytkownika – wykonywane są instrukcje
- w trybie jądra – wykonywane są instrukcje
- gotowy do wykonywania – proces czeka na przydział procesora przez program szeregujący, w tym stanie jest zwykłe wiele procesów
- czekający (uspiony) – proces czeka na wystąpienie zdarzenia do jego dalszego wykonania, na zakończenie operacji `we / wy`
- zakończony – proces zakończył wykonywanie.
- 
- 

#### 14. Co to jest „shell”?

Shell jest to program, który pośredniczy między jądrem, systemem plików i programami użytkowymi. Shell to interpreter poleceń.

#### 15. Podać przykłady programów shell i ich właściwości?

- Przykładowe shelle to shell Korna, shell Bourne’a, a shell C.
- Podstawowe funkcje:
  - Przekazywanie wybranych do programu wybranego polecenia użytkownika,
  - Wykonywanie wbudowanych poleceń,
  - Dostosowywanie języka do pisania skryptów,
  - Ustawianie środowiska pracy,
  - Przywoływanie i edycja uprzednio wydawanych poleceń,
  - Przeadresowanie wejścia – wyjścia poleceń,
  - Generowanie nazw plików,
  - Umieżliwienie przetwarzania w drugim planie.

#### 16. W jaki sposób program shell interpretuje polecenie?

- Interpretacja programu przez shell odbywa się w następującej kolejności:
  - wczytanie polecenia do wewnętrznego bufora
  - podzielenie polecenia na części zwane słowami, shell określa znaczenie każdego słowa,
  - szukanie i przetwarzanie znaków specjalnych
  - zlecenie wykonania polecenia
  - „uspiony” shell czeka na zakończenie polecenia
  - po zakończeniu wykonywania polecenia zgłasza gotowość przyjęcia nowego polecenia.

#### 17. Na czym polega wykonywanie polecenia w tle (w drugim planie)?

Shell zleca wykonanie polecenia i wyprowadza zgłoszenie gotowości przyjęcia następnego.

#### 18. Co to jest planowanie (szeregowanie) procesów?

Planowanie polega na określeniu w jakiej kolejności procesy uzyskują dostęp do zasobów komputera: pamięci operacyjnej. Celem planowania jest zapewnienie jak najszybszego wykorzystania procesora.

#### 19. Podać przykłady algorytmów szeregowania procesów i wyjaśnić ich działanie?

Przykłady algorytmów planowania:

- FIFS (First Come, First Served) – pierwszy nadziedz, pierwszy obskuzny. Realizacja za pomocą kolejki FIFO. Wady: efekt kłopotliwy, kłopotliwy w systemach podziału czasu. Jest algorytmem niewyłączającym.
- SJF (Shortest Job First) – najkrótsze zadania najpierw. Procesor przydzielany jest procesowi o najkrótszej następnej fazy procesora. Teoretycznie daje minimalny czas oczekiwania. Wymaga jednak dokładnego oszacowania czasu przyszłej fazy procesora dla każdego procesu. Szacowanie to wykonywane jest zwykle na podstawie pomiarów czasu faz poprzednich. Może być wyłączone lub nie.
- Planowanie priorytetowe – każdemu procesowi przydzielony jest pewien priorytet. Wybierany jest proces o najwyższym priorytecie. Algorytm SJF jest szczególnym przypadkiem. Może być wyłączającym lub nie.
- Planowanie rotacyjne – zaprojektowany dla systemów systemów podziałem czasu. Ustalony jest kwant czasu (10 – 100 ms). Kolejka procesów ma charakter cykliczny. Kolejnym procesom przydzielany jest co najwyżej kwant czasu.

#### 20. Jak rozwiązuje się zagadnienie planowania procesów w systemach typu UNIX?

Kryteria planowania:

- wykorzystanie procesora w % czasu
- przepływność – liczba procesów kończonych w jednostce czasu
- czas cyklu przetwarzania – średni czas przetwarzania procesu od chwili utworzenia do zakończenia
- czas oczekiwania – średni czas oczekiwania w kolejkach
- czas odpowiedzi – w systemach interakcyjnych czas między zgłoszeniem zamówienia przez użytkownika a pierwszą odpowiedzią

Przykłady algorytmów planowania (szeregowania) są w punkcie 19.

#### 21. Kiedy, po co i jak wykonywany jest proces ładowania systemu?

- Celem procesu ładowania jest umieszczenie systemu operacyjnego w pamięci operacyjnej i rozpoczęcie jego wykonywania. Składa się z kilku etapów:
  - \* inicjalizacja i testowanie sprzętu
  - \* wczytanie i umieszczenie w pamięci bloku systemowego (blok 0 dysku root) z dysku
  - \* program zawarty w bloku systemowym ładuje jądro systemu operacyjnego operacyjnego pliku systemowego (stand/vmunix) do pamięci. Przekazuje sterowanie do pierwszej instrukcji jądra. Program jądra systemu zaczyna ją wykonywać
  - \* wykrywanie i konfiguracja urządzeń, odwołanie się do systemu funkcji systemowej
  - \* przygotowanie środowiska procesu 0. Wykonanie programu systemu jako procesu 0 wykonywanego w trybie jądra. Utworzenie procesów jądra, np. procesów zarządzania pamięcią
  - \* rozwidlenie procesu 0 (wywołanie algorytmu `fork` z jądra). Utworzony proces 1 tworzy kontekst poziomu użytkownika i przechodzi do trybu użytkownika
  - \* proces 1 wywołuje funkcje systemowe `exec` wykonując program `/sbin/init`
  - \* proces int wczytuje plik `/etc/inittab` i rozmatna procesy
  - \* inicjacja wewnętrzna struktury danych jądra, tworzenie np. listy wolnych buforów,
  - i – węzłów, kolejek, inicjacja struktury segmentów i tablicy stron pamięci.
  - \* sprawdzanie głównego głównego pozostałych systemów plików (ewentualnie uruchomienie `fsck`)
  - \* wywołanie są procesy getty monitorujące konsolę i terminale systemu komputerowego, zgodnie z deklaracjami w pliku `/etc/inittab`, a proces `init` wywołuje funkcje systemowe `wait` monitorując zakończenie procesów potomnych. Proces `init` tworzy również procesy demonów.

#### 22. Jaki proces ma PID = 1, jakie zadania wykonuje?

Przedmiotem wszystkich procesów procesów system jest proces o nazwie `init` oraz identyfikatorze PID równym 1, tworzony w czasie startu systemu. Proces `init` odpowiada za powoływanie wielu innych procesów (zawsze na drodze `fork-and-exec`), wśród których wiele wykonuje program `getty`. Każdy z procesów `getty` związane jest z innym terminalem i odpowiada za wyświetlenie na danym terminalu znaku zachęty do logowania oraz rozpoczęcia sesji.

#### 23. Co to są pliki i jakie typy plików występują w systemie UNIX?

Plik to jednostki logiczne przechowywanej informacji, niezależne od właściwości fizycznych urządzeń pamięciowych. Zwykle w plikach przechowywane są programy lub dane. W Unixie, inaczej niż w innych systemach operacyjnych, pojęcie pliku ma kluczowe i centralne znaczenie. Na przykład polecenia są wykonywalnymi plikami, przechowywanymi w góry przewidywalnych miejscach drzewa katalogowego. Systemowe przywileje i prawa dostępu kontrolowane są prawie wyłącznie przez prawa dostępu do odpowiednich plików. Nawet realizacja operacji wejścia / wyjścia do urządzeń i do plików z punktu widzenia użytkownika jest nierozdzielna. Także komunikacja między procesami odbywa się z użyciem struktur reprezentowanych w systemie jako pliki.

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?

Jako i – węzeł rozumiemy będziemy przechowywaną na dysku strukturę, opisującą atrybuty pliku, włączając w to fizyczną lokalizację bloków danych. W momencie przydzielania danej partycji dyskowej potrzebny przechowywany plików, od razu tworzone jest pewne i określa ilość i – węzłów. Stanowi ona limit wyznaczający maksymalną liczbę wszystkich rodzą plików, w tym plików zwykłych, katalogów, plików specjalnych, łączników, które będą mogły być utworzone na tym fragmencie dysku. Standardowo na jeden plik przypada 2 do 8 kB bloków, czyli liczba i – węzłów wyliczone jest automatycznie na podstawie rozmiaru danego fragmentu dysku i w zasadzie jest zupełnie wystarczająca. Wszystkie i – węzły mają unikatowe numery. Z każdym plikiem związany jest opisujący go i – węzeł, rezerwowany w momencie tworzenia danego pliku.

#### 25. Jaką informację są przechowywane w i-węźle?

W i – węźle przechowywane są następujące informacje o plikach:

- identyfikator właściciela i grupy plików
- typ pliku (plik zwykły, katalog itd. lub 0, gdy i – węzeł nie jest używany)
- prawa dostępu do plików
- czas utworzenia, ostatniego dostępu i ostatniej modyfikacji pliku
- czas ostatniej modyfikacji i – węzła
- liczba linków (dowiązani) do pliku
- rozmiar pliku w bajtach
- 12 wskaźników zawierających adresy bloków z danymi pliku ( bloki bezpośrednio adresowane)
- wskaźnik zawierający adres bloku, w którym przechowywane są adresy bloków z adresami bloków z danymi (pośrednio adresowane dwustopniowo)
- wskaźnik wykorzystywany w adresowaniu pośrednim trzypięciowym.

#### 26. Jaką informację przechowywane są w pliku typu "katalog"?

Katalogi służą do powiązania nazw plików z danymi znajdującymi się na dysku. W każdym katalogu może znajdować się pewna liczba plików i innych katalogów (podkatalogów). Katalog jest przechowywany jak plik zwykły i (w ujęciu) ma postać tabeli o dwóch kolumnach. Każdy wiersz tej tabeli zawiera nazwę zainicjującego się w katalogu pliku ( lub podkatalogów) oraz pewien numer, pozwalający na odszukanie atrybutów i danych, które się w nim znajdują.

#### 27. Co to jest katalog z punktu widzenia użytkownika, a co z punktu widzenia budowy systemu plików?

System plików -> lista i-węzłów i odpowiadających im nazw plików

#### 28. Jaka jest różnica między adresowaniem bezpośrednim a adresowaniem pośrednim?

Do adresowania bezpośredniego służą pierwsze 12 wskaźników i – węzła. W każdym z tych wskaźników znajduje się adres bloków z danymi. Krótko mówiąc każdy z tych 12 wskaźników wskazuje bezpośrednio na blok danych. Adresowanie bezpośrednie pozwala na zaadresowanie danych plików o rozmiarach nieprzekraczających 48 kB. Aby móc zaadresować plik o rozmiarze większym niż 48 kB należy użyć adresowania pośredniego. Do adresowania pośredniego służą wskaźniki 13, 14, 15 i – węzła. Każdy z tych wskaźników nie wskazuje bezpośrednio na blok danych tylko na blok (bloki), w którym znajdują się dopiero adresy bloków danych i tak mamy odpowiednio:

- 13 wskaźnik zawiera adres bloku, w którym umieszczone są adresy bloków z danymi o rozmiarze nie przekraczającym:  $48kB + 1024^4kB = 4144kB$
- 14 wskaźnik to adres bloku, w którym umieszczone są adresy bloków zawierających adresy bloków z danymi o rozmiarze nie przekraczającym:  $48kB + 1024^4kB + 1024^4kB = 4198448kB$
- 15 wskaźnik to adres bloku, w którym umieszczone są adresy bloków na adresy następnych bloków zawierających adresy bloków z danymi o rozmiarze nie przekraczającym:  $48k + 1024^4kB + 1024^4kB + 1024^4kB = 4299165744kB$

#### 29. Wyjaśnić mechanizm rozmieszczania bloków i fragmentów pliku w blokach na dysku?

Do przechowywania danych na dysku system UNIX używa bloków i fragmentów. Fragment jest najmniejszą jednostką przestrzeni dyskowej zajmowanej przez plik. Rozmiar bloku jest całkowitą wielokrotnością rozmiaru fragmentu (stosunek rozmiaru bloku do rozmiaru fragmentu nie może jednak przekraczać 8). Na przykład, rozmiar bloku wynosi 8kB a rozmiar fragmentu 2kB.

#### 30. Jak adresowane są bloki i fragmenty pliku na dysku?

- Reguły przydziału bloków i fragmentów:
  - Jeśli rozmiar jest mniejszy niż rozmiar fragmentu, plikowi temu przydzielony jest pierwszy wolny fragment.
  - Jeżeli rozmiar pliku jest większy niż rozmiar fragmentu, ale mniejszy niż rozmiar bloku, plikowi temu przydzielane są kolejne fragmenty należące do tego samego bloku.
  - Jeśli plik jest większy niż rozmiar bloku, plikowi temu przydzielana jest odpowiednia liczba bloków, niekoniecznie znajdujących się obok siebie, o łącznym rozmiarze nieprzekraczającym rozmiaru pliku. Pozostała część pliku umieszczona jest w pokonanych obok siebie fragmentach należących do jednego bloku zgodnie z regułami a) oraz b).

#### 31. Czym różni się przydział ciągły miejsca na dysku od przydziału listowego?

Przydział ciągły: Plik zajmuje ciąg kolejnych bloków na końcu jednego jest adres następnego. Przydział listowy: Plik stanowi listę powiązanych bloków które mogą znajdować się w dowolnym miejscu na dysku.

#### 32. Co jest tablica FAT?

FAT: Tablica opisująca, w których kłastrach dysku twardego lub dyskieci magnetycznej system operacyjny ma szukać każdego z zapisanych na nim plików. FAT jest tworzony podczas formatowania nośnika danych. Podczas zapisu pliku informacje o nim są automatycznie zachowywane w tablicy FAT. 2. Polecenie `ls` przez FAT rozumie się FAT16 lub FAT32. Od ang. File Allocation Table - tablica alokacji plików.

#### 33. Porównać przydział listowy miejsca na dysku z przydziałem indeksowym?

Przydział listowy -> lista powiązanych bloków na końcu jednego jest adres następnego. Jeśli potrzeba dostęp sekwencyjny. Przydział indeksowy -> blok indeksu z numerem zawierający listę poszczególnych bloków na dysku dostęp bezpośredni

#### 34. Co to jest i jak jest wykorzystywana pamięć operacyjna?

Pamięć, do której komputer ładuje aktualnie używane dane, tak, aby były one błyskawicznie dostępne dla procesora. RAM jest dużo szybszy (tzn. ma znacznie krótszy czas dostępu, liczoney w nanosekundach) od pamięci masowych, takich jak dyski twarde, napędy CD-ROM czy napędy dyskieci. Jednak w przeciwieństwie do tych typów pamięci dane zawarte w RAM giną po wyłączeniu komputera - a więc RAM do pracy wymaga stałego źródła zasilania.

#### 35. Wyjaśnić mechanizm wymiany (swapping) procesów?

Wymiana (swapping) jest to mechanizm, za pomocą którego UNIX rozdziela dostępną pamięć między aktualnie aktywne procesy, gdy ich łączne wymagania pamięciowe przekraczają fizyczną pamięć operacyjną. Swapowanie polega na zapisaniu na dysku całego procesu, a tym samym zwolnieniu całej związanej z nim pamięci. Proces przeniesiony do obszaru `swap` dysku musi zostać ponownie wczytany do pamięci, by można było go kontynuować.

#### 36. Wyjaśnić mechanizm stronicowania na żądanie?

Stronicowanie (paging) wymaga przenoszenia części pamięci przydzielonej procesowi – w jednostkach zwanych stronami (page) – na dysk, aby zwinąć pamięć potrzebną dla tego bądź innego procesu. Stronicowanie na żądanie wykorzystuje zasadę kolejkowania i do pamięci przesyłane są strony niezbędne w danej chwili lub, które niebawem mogą się okazać niezbędne. Strony mogą znajdować się na dysku, w pamięci głównej lub w obszarze wymiany. Jeśli niezbędna jest strona znajdująca się na dysku, generowany jest błąd strony i odpowiednie przerwanie. Wykonywanie procesu jest wstrzymywane, po czym odnadywana jest wolna ramka, do której przypisywana jest potrzebna strona.

#### 37. Wyjaśnić mechanizm segmentacji?

Segmenty to określone fragmenty programu, np. program główny, podprogram i biblioteki, funkcje, symbole, dane, stos. Do podstawowych zalet segmentacji należą: możliwość wydzielenia i konfiguracji urządzeń, ochrony przed nieuprawnionymi dostępami, wydzielenia segmentów przez różne procesy. Na przykład ustawienie bity 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 ujęciu, segmenty danych i stosu oraz wskaźniki do właściwego miejsca w segmentie programu. Aby odwołać się do odpowiedniego segmentu, procesy korzystają z tablicy segmentów. Każdy element tej tablicy można przedstawić w postaci pary, adres bazowy danego segmentu oraz jej rozmiar. Tablicę segmentów przechowuje się zwykle w pamięci głównej, a tylko pewną liczbę elementów tej tablicy przechowuje się w rejestrach.

#### 38. Podać przykłady algorytmów wymiany stron?

Do algorytmów wymiany stron zaliczamy algorytmy:

- FIFO
- Algorytm optymalny
- LRU (Least Recently Used)

#### 39. Kiedy występuje błąd strony i jak jest obsługiwany?

Błąd strony (page fault) pojawia się wtedy, gdy procesowi potrzebna jest strona pamięci, która rezyduje w fizycznej pamięci operacyjnej i musi zostać wczytana z dysku.

#### 40. Na czym polega zarządzanie pamięcią wykonywane przez system operacyjny?

Tablica bajtów z jednoznacznym adresem. Do pamięci wpada adres z informacją czy zapisać czy odczytać z pod danego adresu. Przydział pamięci -> metoda nakładek, dzielenie na elementy rozłączne, wykonywanej wymiana, pamięć umieszcza części procesów na dysku i pobiera kiedy chce; stronicowanie, pamięć rocesu dzielona na stałe bloki zwane stronicami

#### 41. Jak są zadania podsystemu wejście - wyjście?

Interfejs przestrzeni nazw plików, nadzoru przebieg dostępu do plików urządzeń, nadzoru poprawności operacji, przydziela miejsca w systemie, urządzenia, planuje operacje wejścia wyjścia, buforowanie, pamięć rocesu dzielona na stałe bloki zwane stronicami

#### 42. Co to są pliki specjalne i do czego służą?

Pliki specjalne w UNIXie umożliwiają operacje wejścia / wyjścia do urządzeń. Pliki te przechowywane są standardowo w katalogu `/dev` (raz w jego podkatalogach – System V). W plikach specjalnych nie przechowuje się żadnych danych. Pliki te charakteryzują sposób działania urządzenia, wskazując miejsce podłączenia urządzeń do systemu oraz zapewniając dostęp programów obsługi urządzeń.

#### 43. Co to jest tablica rozdzielcza urządzeń we-wy, jakie informacje zawiera i do czego służy?

Interfejs między systemem operacyjnym a programami obsługi urządzeń. Tablica rozdzielcza urządzeń blokowych -> `open_dev`, `close_dev`, `strategia_dev` - transmisja danych między buferem a urządzeniem. Tablica rozdzielcza urządzeń znakovych -> `open_rdev`, `close_rdev`, `read_rdev`, `write_rdev`, `ioctl_rdev` - input output control

#### 45. Jak są typy plików specjalnych?

- Pliki specjalne znakowe
- Pliki specjalne blokowe

#### 46. Jaką informację są zapisane w plikach specjalnych i do czego służą?

Pliki specjalne odpowiadają za realizację znakową, czyli surowego dostępu do urządzeń. Pliki specjalne blokowe umożliwiają dostęp do bloków. Pliki specjalne znakowe używane są w celu wykonania niebuforowanego przesyłania danych na i z urządzenia, podczas gdy pliki specjalne blokowe wykorzystywane są wtedy, gdy dane mają być zapisywane bądź odczytywane porcjami, określającymi blok bloki.

#### 47. Jaką rolę pełnią podprogramy obsługi urządzeń?

Podprogram obsługi urządzenia jest to taki zbiór funkcji z jądra systemu, do którego dostęp otrzymuje się poprzez tablicę rozdzielczą urządzeń.

Podprogramy obsługi urządzeń mają jednak pewną cechę, która w dość naturalny sposób wyróżnia je spośród innych części jądra systemu. Cechą tą jest metoda dostępu do ich funkcji *sposób* jądra. Normalnie UNIX udostępnia użytkownikowi swoje usługi poprzez mechanizm funkcji systemowych. Tymczasem dostęp do podprogramów obsługi urządzeń otrzymuje się przez system plików.

Termin „podprogram obsługi urządzenia” pochodzi stąd, że często funkcja przez ten podprogram udostępniana są to funkcje pozwalające użytkownikowi operacje wejścia-wyjścia na fizycznych urządzeniach, takich jak dysk twarły lub drukarka.

#### 48. Co to jest pamięć współdzielona?

W jej modelu procesy posługują się systemowymi funkcjami odwzorowania pamięci, aby uzyskać dostęp do obszarów pamięci należących do innych procesów. System operacyjny na ogół próbuje zapobiegać dostawieniu się jednego procesu do pamięci innego.

#### 49. Co są funkcje systemowe?

Funkcje systemowe tworzą interfejs między programem a systemem operacyjnym. Można podzielić je na kilka kategorii: nadzorowanie procesów, operacje na plikach, operacje na urządzeniach, komunikacja...

#### 44. Jak wykorzystywana jest pamięć podręczna do wydajniejszego korzystania z systemów dyskowych?

Niektóre sterowniki dysków mają pamięć podręczną, używają do przesyłania danych między systemem a sterownikiem i dyskiem. Przechowuje się tu ostatnio odczytane, zapisane dane. Gdy są jakies dane potrzebne najpierw system szuka ich w pamięci podręcznej.