* **Кај континуирано доделување на меморија,бит мапите претставуваат**

Динамична алокација на мемориите

* **Под кои услови се појавува застој?**
* **Кои се програмски закани за системот?**

**-**Тројански коњи , Замка и Преполнување на стек или бафер

* **Опишете го проширениот век на процесорот:**

Процесот се состои од низа чекори кои што следат еден по друг. Помеѓу два чекори може да биди прекинат, а неговото извршување да продолжи во друг момент на ист или друг процесор. Носењето на процесот од една во друга состојба го врши оперативниот систем. Процесот може да се најде во неколку состојби ( 5 или 7 зависно од конечниот автомат кој што е користен за опишување на состојбата), а следните три се најважни: Состојба на извршување (RUN, RUNNING) процесорот извршува инструкции на овој процес; Состојба на чекање на процесор (READY,RUNNABLE) процесот ги добил сите потребни ресурси освен процесорот и спремен е за работа и да му се додели процесор; Состојба на чекање на ресурс (WAIT,UNRUNNABLE) процесот чека на некое случување (на пример да се изврши печатење) бидејќи за извршување на процесот е потребен ресурс ( во состојба WAIT чека извршување на влезно-излезните операции или резултат на некои други процеси).

* **Под кои услови се појавува застој:**

1. Линеарно чекање
2. При користење на ресурсите се почитува принципот на меѓусебно исклучуавње
3. Кружно чекање
4. Доделување на ресурсите се обавува без претпразнење-на процесот не може да му се одземе ресурс
5. Процесот здружува еден ресурс бидејќи му е потребен и чека за ресурс кој го користи некој друг процес
6. Доделување на ресурсите се обавува со претпразнење- на процесот не може да му одземе ресурс
7. При користењето на ресурсите не се почитува принципот на меѓусебно исклучување

* **Во што е разликата помеѓу распоредување без предпразнење и распоредување со предпразнење?**

Распоредување со претпразнење: Процесорот може да се одземе од процес кој не завршил со своите активности, неопходна хардверска компонента за реализација е тајмер кој поставува прекинувачки сигнал, претпразнењето влијае врз конструкцијата на јадрото -Распоредување без претпразнење: Процесот може да се одземе само од процес кој ги завршил своите активности или чека некој ресурс.

* **За да јадрото ја реализира својата функција неопходно е самиот хардвер да исполни одредените предуслови, односно да поседува одредени компоненти. Кои се тие компоненти?**

**Механизам за прекинување – обезбедува извршување на управувачката програма (прекинувачки способности**) т.е прекинување на контролите за извршување на корисникот и на управувачката програма. Најмалку што механизмот треба да направи е:

**\*да ја сочува вредноста на програмскиот бројач на прекинатата корисничка програма**

**\*да ја покрене управувачката програма** со фиксна локација во меморијата

**\***управувачката програма понатаму одредува извор на прекини и реагира на соодветен начин

**Заштитен механизам за адресирање на меморија – спречува погрешно адресирање**

**\***т.е спречува еден процес да ги запиши следните податоци во делот на меморијата кој е во другиот процес;

**\***овој процес е автоматски и ги чува и интегрира процесот и податоците кои се наоѓаат во работната меморија;

**Збир на привилегирани инструкции** – сите инструкции кои се достапни на ОС, но не и на корисничката програма;

\*овие инструкции овозможуваат оперативниот систем да ги сокрива прекините;

\*на процесорот му доделува друг процес;

\*пристапува на регистрите во меморијата

\*извршува влезни-излезни операции;

\*за разлика од непривилегираната инструкција која се извршува во корисничкиот режим, при извршување на привилегирана инструкција оперативниот систем се наоѓа во системскиот режим;

\*корисничката програма (КП) не може да извршува привилегирана инструкција директно, туку само со системски повик;

\*КП со системскиот повик бара од ОС да изврши привилегирана инструкција после што ОС преминува во системскиот режим и ја извршува таа инструкција;

**Часовник за реално време (real-time clock**) – помага сатниот механизам на реалното време да ги контролира и да ги евидентира потрошените ресурси на компјутерот за сите поединечни процеси.  
– Овај механизам може да се користи и за распоредување и закажување и извршување на различна работа.

* Една од функциите на планерот на работи е?

1. Ги дели работите на процеси
2. Извршува суспендирање на процесот

* **Контролерот работи на три начини. Графичката картичка е:**
* **Што претставува MMU**

MMU (Memory-Menagement Unit) – единица за управување со меморија.

* **Софтвер со кој се пресретнуваат TCP/IP пакети кои поминуваат низ одреден компјутер и по потреба се прегледува нивната содржина претставува:**

1. Sniffing
2. Смтроносен ping
3. Smurf
4. dos

* **Што е контролен блок на процес?**

1. Дел од меморијата односно мемориска структура со основни информации за процесот кој што OC го користи за управување со тој процес
2. Дел од оперативниот систем кој овозможува контрола, извршување и управување со процесите

* **Кои се дистрибуирани OC? (може да изберете повеќе опции)**

1. Систем во кој секој компјутер има свој оперативен систем
2. Систем во кој меѓусебно можи да се разменуваат податоци со помош на одредени протоколи
3. Систем во кој компјутерите содржат релативно висок степен на автономија
4. Систем кој освен споделување и миграција на датотеки овозможува и споделување на процеси т.е програми

* **Дали екстерната фрагментација е карактеристика за континуирана или дисконтинуирана алокација на меморијата?**

дисконтинуирана алокација на меморија

* **Контролниот регистер кај типичен влезно-излезен приклучок**

1. Служи за поставување режим на работата на уредот
2. Служи за да покажува дали командата е извршена
3. Служи за запишување на податоци на излениот уред

* **Во што е разликата помеѓу распоредување без предпразнење и распоредување со препразнење?**

1. Во случај на распоредување со претпразнење, процесорот може да му се одземе на процесот кој не ги завршил своите активности. Во случај на распоредување без препазнење, тоа не е можно
2. Во случај на распоредување со претпразнење, процесорот може да му се доделува на процес со највисок приоритет. Во случај на распоредување без претпразнење тоа не е можно
3. Во случај на распоредување без претпразнење, процесорот може да му се одземе на процесот кој не ги завршил своите активности. Во случај на распоредување со претпразнење, тоа не е можно

* **Кои се програмски закани за системот? (изберете повеќе опции)**

1. Лажирање (spoofing) на DNS
2. Тројански коњи
3. Замка (trap door)
4. Преполнување на стекот или баферот
5. Smurf

* **Кеш меморијата се користи за**

1. Област на брза системска меморија која ја чува копијата на податоците
2. Да чува било која копија од дискот
3. Да ги чува актуелните податоци
4. Да се избегне програмирањето на ВИ за трансфер на големи количини на податоци

* **Кои се предностите од употреба на програми со повеќе нишки?**

1. Спречување програмата да влези во deadlock
2. Подобро искористување на повеќепроцесорската архитектура
3. Намалување на времето на одзив
4. Поефикасно делење на ресурсите и економичност

* **Кај континуирано доделување на меморијата, бит мапите претставуваат**

1. Динамичка алокација на меморијата
2. Статичка алокација на меморијата
3. Статичко поврзување на меморијата
4. Динамичко поврзување на меморијата

* **Кај континуирано доделување на меморијата, системот на здружени парови- другари (buddy system) претставува**

1. Динамичка алокација на меморијата
2. Динамичко поврзување на меморијата
3. Статичка алокација на меморијата
4. Статичко поврзување на меморијата

* **Дали екстерната фрагментација е карактеристика за:**

1. Континуирана алокација на меморијата
2. Дисконтинуирана алокација на меморијата
3. Двата вида алокација на меморијата

* **Стопирање на работата на сервисот или програмата така што на другите корисници им се оневозможува да работат со нив претставува:**

1. Смртоносен ping
2. Dos
3. Smurf
4. Sniffing

* **Бришење на сите фајлови од фолдерот c:\sistemski**

Echo 1. Промена на наслов

Echo 2. Печатење со лооп

Echo 3. Бришење на фајлови

* **Нема прашање**

1. Пуштање на пораки
2. Користење делливи мемории
3. Изменувани цевки

* **На кои се начини може да се изврши идентификација на корисникот на OC?**

Генерално идентификацијата може да се оствари на три начини:

-специјален хардвер како што е клуч или ID картичка

-внесување на доверливи информации како што се лозинката

-биолошки атрибути на корисникот (отпечаток на прстите, потпис, снимање на очна рожница итн.)

* **DMA се користи за**

1. Да се избегне програмирањето на ВИ за трансфер на големи количини на податоци
2. Да чува било која копија од дискот
3. Област на брза системска меморија која ја чува копијата на податоците
4. Да ги чува актуелните податоци

* **Дали е можно системо да се најде во состојва на застој до колку на него постои само еден процес?**

1. Не е можно. Ова е директна последица на условите на задржување на ресурсите и чекање на други ресурси
2. Можно е. Ова е директна последица на условите на задржување на ресурсите и чекање на други ресурси

* **Што претставува интерна фрагментација?**

1. Просторот од меморијата што останува неискористен меѓу партициите
2. Просторот од меморијата што останува неискористен во партициите
3. Просторот од меморијата што останува неискористен меѓу партициите на почетокот на доделувањето

* **Статусниот регистер кај типичен влезно-излезен приклучок**

1. Служи за поставување режим на работата на уредот
2. Служи за запишување на податоците на излезниот уред
3. Служи за да покажува дали командата е извршена

* **Една од функциите на средниот распоредувач е да**

1. Доделува процесор на процесите кои што се наоѓаат во процесорскиот ред
2. Ги дели работите на процеси
3. Извршува суспендирање на процесот

* **Контролерот работи на три начини. Серискиот контролер е**
* **Што претставува дел од јадрото на OC?**

1. Планер за работи од ниско ниво
2. Семафор
3. Планер за работи од средно ниво
4. Именувани цевки
5. Планер за работи од висок ниво
6. Рутини за обработка на прекини

* **Која е задачата на диспечерот?**

1. Бира процес од редот на чекање
2. Доделува процесор на процесите кои што се наоѓаат во процесорскиот ред
3. Извршува суспендирање на процесот
4. Кеш меморијата се користи за : да чува било која копија од дискот
5. Кои се дистрибуирани ОС
6. Контролнио регистар кај типичен влезно-излезен приклучок

**\*Регистер на податоци о влезен режим**

**-**Влезниот регистер служи за читање на податоци од влезните уреди. Процесорот од влезниот регистер исклучиво ги чита податоците

**\*Регистер податоци за излезен режим**

-Излезниот регистер служи за запишување податоци на излезниот уред . Од овој регистер процесорот исклучиво чита информации.

1. Што е контролен блок на процес?

- Контролниот блок е дел од меморијата односно мемориска структура со основни информации за процесот кој што ОС ги користи за управување со тој процес.

1. Софтвер со кој се пресретнуваат TCP/IP пакетите кои поминуваат низ одреден компјутер и по потреба се прегледува нивната содржина претставува **(sniffer).**
2. Кај континуирано доделување на меморија,бит мапите претставуваат **статича алокација на меморијата\*\*\*\*\*\*\***
3. Кои се програмси закани за системот? **Тројански коњи , Замка и Преполнување на стек или бафер.**
4. Што претставува ММU.

-Мапирањето (пресликувањето) на виртуелниот адресен простор во физички го прави хардверскиот уред кој се вика MMU (Memory-Menagement Unit) – единица за управување со меморија.

1. Под кои услови се појавуа застој?
2. Дали екстерната фрагментација е карактеристика за континуирана или дисконтинуирана алокација на меморијата? **дисконтинуирана алокација на меморија**
3. Кои се предностите од употреба на програми со повеќе нишки?

Предноста од употреба на програми со повеќе нитки се:

-намалување на времето на одѕив - повеќенитна техника која им овозможува на интерактивните апликации да продолжат со работа дури и кога дел од програмата е блокиран или извршува некоја долготрајна апликација.

-поефикасно делење на ресурсите и економичност – на пример делење на сегменти на кодот овозможува голем број на нитки од една апликација да се извршува во ист адресен простор. Исто така изработка на процеси и префрлување на контекстот се поспори од генерирање на нитки и префрлување на контекстот на нитки.

-Подобро искористување на повеќепроцесорската архитектура – било кои нитки може да се извршуваат истовремено, секоја на различен процесор.

1. Опиши го проширениот животен век на процесорт?

- Процесот се состои од низа чекори кои што следат еден по друг. Помеѓу два чекори може да биди прекинат, а неговото извршување да продолжи во друг момент на ист или друг процесор. Носењето на процесот од една во друга состојба го врши оперативниот систем. Процесот може да се најде во неколку состојби ( 5 или 7 зависно од конечниот автомат кој што е користен за опишување на состојбата), а следните три се најважни: ¬ Состојба на извршување (RUN, RUNNING) процесорот извршува инструкции на овој процес; ¬ Состојба на чекање на процесор (READY,RUNNABLE) процесот ги добил сите потребни ресурси освен процесорот и спремен е за работа и да му се додели процесор; ¬ Состојба на чекање на ресурс (WAIT,UNRUNNABLE) процесот чека на некое случување (на пример да се изврши печатење) бидејќи за извршување на процесот е потребен ресурс ( во состојба WAIT чека извршување на влезно-излезните операции или резултат на некои други процеси).

1. една од функциите на планерот е ги дели работите на процеси
2. Како работи контролерот? Контролерот работи на три начини -како приклучок (Port) -како магистрала (Bus) -како уред (device)
3. Механизам за прекинување –обезбедува извршување на управувачката програма (прекинувачки способности) т.е прекинување на контролите за извршување на корисникот и на управувачката програма. Најмалку што механизмот треба да направи е: \*да ја сочува вредноста на програмскиот бројач на прекинатата корисничка програма \*да ја покрене управувачката програма со фиксна локација во меморијата \*управувачката програма понатаму одредува извор на прекини и реагира на соодветен начин ¬ Заштитен механизам за адресирање на меморија –спречува погрешно адресирање \*т.е спречува еден процес да ги запиши следните податоци во делот на меморијата кој е во другиот процес; \*овој процес е автоматски и ги чува и интегрира процесот и податоците кои се наоѓаат во работната меморија; ¬ Збир на привилегирани инструкции –сите инструкции кои се достапни на ОС, но не и на корисничката програма; \*овие инструкции овозможуваат оперативниот систем да ги сокрива прекините; \*на процесорот му доделува друг процес; \*пристапува на регистрите во меморијата \*извршува влезни-излезни операции; \*за разлика од непривилегираната инструкција која се извршува во корисничкиот режим, при извршување на привилегирана инструкција оперативниот систем се наоѓа во системскиот режим; \*корисничката програма (КП) не може да извршува привилегирана инструкција директно, туку само со системски повик; \*КП со системскиот повик бара од ОС да изврши привилегирана инструкција после што ОС преминува во системскиот режим и ја извршува таа инструкција; ¬ Часовник за реално време (real-time clock) –помага сатниот механизам на реалното време да ги контролира и да ги евидентира потрошените ресурси на компјутерот за сите поединечни процеси. –Овај механизам може да се користи и за распоредување и закажување и извршување на различна работа.

Системски софтвер група 1 14-12-2016

1. Што вклучува процесот? Објаснете го проширениот дијаграм на состојби на процесите!

Процесот е програма во состојба на извршување и покрај програмскиот код и двете фундаментални мемориски секции(stack секција и секција на податоци со глобални променливи) тој вклучува:

* вредност на програмскиот бројач
* вредност на останатитите важни регистри на процесорот
* влезно-излезни ресурси кои што процесор ги користи
* контролен блок на процесот
* опишувач на процесор(process descriptor)

Основниот дијаграм на состојби на процесите се состои од 5 состојби и може да се прикаже со конечен автомат. 5-те состојби се: Start, Stop, Ready, Wait, Run.

Во реалните оперативни системи постои можност процесот да се прекине со работа привремено. Во таквите оперативни системи тоа може да го направи корисник човек или самиот оперативен систем. Таквото прекинување на процесот се нарекува суспендирање што значи дека треба да се воведи проширен дијаграм на состојби.

Во проширениот дијаграм на состојби се воведуваат две нови состојби: Suspend Ready и Suspend Wait од причина што суспендирање на процесот може да се направи само кога процесот е во состојба Ready или состојба Wait.

Новите транзиции се:

Wait → Suspend Wait се доведува процесот од состојба Wait во состојба Suspend Wait

Suspend Wait → Suspend Ready ресурсот неопходен за понатамошно извршување на процесот е слободен, но процесот е и понатаму суспендиран

Suspend Wait → Wait процесот се ослободува од суспендирање но ресурсот кој е неопходен за процесот не е слободен. Оваа транзиција е можна на експлицитно барање на корисникот човек.

Suspend Wait → Wait процесот се ослободува од суспендирање и се става во редот за чекање процес. Оваа транзиција е можна на експлицитно барање на корисникот човек.

2. Која е задачата и како функционира диспечерот?

Задачата на диспечерот е да доделува процесор на процесите кои се во состојба Ready и се наоѓаат во процесорскиот ред. За разлика од планерот на работи, диспечерот работи со процесите кои се во редот на чекање за процесор додека планерот се повикува кога ќе се појават нови процеси или завршат постоечки.

Диспечерот прво одредува дали процесорот треба да продолжи со извршување на процесот или да му се додели на друг процес.

* Ако продолжува тековниот процес, контролаа се враќа на адресата зачувана во прекинувачкиот механизам на јадрото
* Ако процесорот се даде на друг процес, диспечерот прво го ажурира контекстот на последниот процес што се извршуваше, па врз основа на контекстот на новиот процес поставува окружување во кое тој процес може да се изврши, потоа контролата на извршување се префрла на она место во новиот процес каде што извршувањето било прекинато.

3. Објаснете ги следните алгоритми за доделување на меморијата:

а) first-fit, б) best-fit, в) worst-fit, г) next-fit.

Како може да се оправда користењето на worst-fit алгоритмот за доделување на меморија?

Што е негова потенцијална предност во однос на best-fit?

a) На процесот му се доделува првата празнина која е доволно голема.

б) На процесот му се доделува најмалата празнина која е доволно голема за да се смести процесот, се пребарува цела листа, ако листата не е сортирана.

в) На процесот му се доделува најголема празнина. Исто како кај best-fit, се пребарува цела листа.

г) Се пребарува од таму каде што застана претходното пребарување. Работи полошо од first-fit затоа што процесите ќе ги сместува во големи празнини од причина што тие празнини се креираат најчесто на крајот од меморијата.

Користењето на worst-fit може да се оправда со тоа што ја намалува екстерналната фрагментација. Исто така, worst-fit ги создава најголемите неискористени делови од празнините во кој се сместува процесот. Овие големи делови би биле повеќе корисни од колку малите неискористени делови кои се појавуваат со best-fit.

4. На кои начини се подобрува ефикасноста на влезно-излезниот потсистем? Што претставува кеширање? Во што е разликата помеѓу кеш и бафер?

За да се подобри ефикасноста на влезно-излезниот потсистем, оперативниот систем мора да обезбеди конкурентност во работата со уредите и различни механизми на кеширање, баферирање, и користење на spooler-и.

Кеширањето е техника на подобрување на преносот на податоци од главната меморија до процесор. Кеш претставуа област на брза мала системска меморија која најчесто мал дел од главната меморија или од hard-disk-от.

И баферите се мали мемории кои најчесто се присотни кај останатие влезно-излезни уреди и додека кешот чува било кој дел од главната меморија или hard-disk-от, баферот ги чува актуелните податоци кои треба да се пренесат кон уредот или кон главната меморија на компјутерот. Баферот има примена во облик на spooler. Предноста на користење spooler е во тоа што процесите не мора да чекаат уредот да се одлободи за да пуштат барање за мемориска комуникација со дадениот уред.

5. Да се напише batch скрипта која во зависност од изборот на корисникот ќе ги извршува слениве акции:

1) промена на наслов во “ФИКТ - Битола”

2) печатење – looping низа на броеви од 30 до 15 со чекор -3

3) бришење на сите фајлови од фолдерот c:\sistemski

|  |
| --- |
| @echo off  echo 1. Promena na naslov  echo 2. Pecatenje so loop  echo 3. Brisenje na fajlovi  set /p izbor="Izbor: "  if %izbor%==1 (  title FIKT-Bitola  ) else if %izbor% ==2 (  for /L %%n in (30,-3,15) do (  echo %%n )  ) else (  del c:\sistemski\\*  ) |

6. Да се најде времето на извршување на секој од процесите P1, P2, P3, и P4 и просечното време на чекање по процес според табелата, ако процесите се извршуваат според FCFS, SJF with preemtion(со претпразнење) и RR алгоритмите. Циклусот за RR трае 4 временски единици. Според кој од алгоритмите процесот P2 ќе заврши најбрзо.

|  |  |  |
| --- | --- | --- |
| Процес | Време на пристигнување | Време на извршување |
| Р1 | 0 | 4 |
| Р2 | 1 | 5 |
| Р3 | 4 | 3 |
| Р4 | 6 | 4 |

FSFS:

|  |  |  |  |
| --- | --- | --- | --- |
| P1 | P2 | P3 | P4 |

0 4 9 12 16

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Процес | Време на пристигнување | Време на извршување | **Време на постоење** | **Време на чекање** |
| Р1 | 0 | 4 | **4-0=4** | **4-4=0** |
| Р2 | 1 | 5 | **9-1=8** | **8-5=3** |
| Р3 | 4 | 3 | **12-4=8** | **8-3=5** |
| Р4 | 6 | 4 | **16-6-10** | **10-4=6** |

Времето на извршување(или постоење) на процесите е дадено во табелата.

Просечно време на чекање: (0 + 3 + 5 + 6) / 4 = 14/4 = 3.5

SFJ with preemtion:

|  |  |  |  |
| --- | --- | --- | --- |
| P1 | P3 | P4 | P2 |

0 4 7 11 16

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Процес | Време на пристигнување | Време на извршување | **Време на постоење** | **Време на чекање** |
| Р1 | 0 | 4 | **4-0=4** | **4-4=0** |
| Р2 | 1 | 5 | **16-1=15** | **15-5=10** |
| Р3 | 4 | 3 | **7-4=3** | **3-3=0** |
| Р4 | 6 | 4 | **11-6=5** | **5-4=1** |

Времето на извршување(или постоење) на процесите е дадено во табелата.

Просечно време на чекање: (0 + 10 + 0 + 1) / 4 = 11/4 = 2.75

RR TQ = 4:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| P1 | P2 | P3 | P4 | P2 |

0 4 8 11 15 16

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Процес | Време на пристигнување | Време на извршување | **Време на постоење** | **Време на чекање** |
| Р1 | 0 | 4 | **4-0=4** | **4-4=0** |
| Р2 | 1 | 5 | **16-1=15** | **15-5=10** |
| Р3 | 4 | 3 | **11-4=7** | **7-3=4** |
| Р4 | 6 | 4 | **15-6=9** | **9-4=5** |

Времето на извршување(или постоење) на процесите е дадено во табелата.

Просечно време на чекање: (0 + 10 + 4 + 5) / 4 = 11/4 = 4.75

Според гантограмите, процесот P2 завршува најбрзо во RR алгоритмот(во временската единица 8!)

Според вкупното време на постоење P2 постоел најмалку во FSFS(8 временски единици!)

Системски софтвер група 2 13-12-2017

1. Што претставува замена на контекстот на процесот?

Процесот, како програма во фаза на избршување може да биде стопиран од работа, и да премини во друга состојба кај мултипрограмските оперативни системи. Тоа значи дека оперативниот систем за секој процес треба да има информации за неговата состојба, негов приоритет како и контекст (опкружување) на тој процес.

Контекстот на процесот уште се нарекува хардверски контролен блок на процесот.

Потатоците во контексот ги генерира оперативниот систем кога се одзема процесорот од процесот, со цел да се овозможи продолжување на извршувањето на процесите.

Замената на контекстот претставува чиста загуба на време, но бидејќи се спроведува мултипрограмирање кога се прекинува одреден процес и се стартува друг, замената на контекстот мора да се изврши.

2. Што претставува socket?

Socket(приклучок) е најчесто користен механизам на размена на пораки преку мрежа за процеси кои се извршуваат на различни компјутери кај клиен-сервер системи.

Претставува крајна точка на комуникацијата.

Парот процеси кој сака да комуницира преку мрежа формира проклучоци на секоја страна на мрежата, по еден за секој процес.

3. Што претставуваат поврзани листи кај алокација на меморијата? Објаснете!

Кај динамичката алогација меморијата се состои од процеси и празнини а оперативниот систем динамички води евиденција за зафатената меморија. Еден начин на водење е виденција за зафатената меморија е со помош на поврзани листи. Поврзаните листи се градат со записи, каде секој запис има одредени полиња како: **Тип на меморија**(дали е зафатена од процес), **Почетна адреса** за делот од меморијата кој ја опишува дадениот запис, **Должина на опишаната меморија**, и **покажувач кон следниот запис**.

4. Што претставува домен на заштита? Набројте неколку видови на напади на системот и објаснете три од нив!

Доменот на заштита е колекција од објекти и операции кои може да ги извршува даден процес ако е припадник на дадениот домен(во суштина, права). Секој домен дефинира множество на објекти и сите операции кои можат да се извршат на тоа множество.

Ако еден процес припаѓа на некој домен, тоа значи дека тој може да ги изврши операциите над објектите зададени во тој домен.

Неколку видови напади на оперативниот систем:

Лажирање со DNS, напаѓачот лажно се претставува како друг компјутер со IP адреси. Може да изврши напад на DNS сервисот давајќи му погрешни информации.

Smurf: Напаѓачот им праќа на сите членови на мрежата ICMP echo-request пакети и како одредишна адреса ја става адресата на жртвата а жртвата приме голем број на echo-reply пакети.

Смртоносен ping: Праќање на голем број на ICMP request пакети кои можат да бидат фатални за некои оперативни системи.

Нушкање, скенирање на приклучоци

5. Да се напише batch скрипта која во зависност од изборот на корисникот ќе ги извршува слениве акции:

а) промена на боја на текст и позадина(по ваш избор)

б) копирање на содржината на фолдерот [c:\studenti](file:///c:\studenti) во фолдерот d:\sistemski

б) печатење на произволна вредност внесена од корисникот.

|  |
| --- |
| @echo off  echo 1. Promena na naslov  echo 2. Pecatenje so loop  echo 3. Brisenje na fajlovi  set /p izbor="Izbor: "  if %izbor%==1 (  color fc  ) else if %izbor% ==2 (  copy [c:\studenti](file:///c:\studenti) d:\sistemski  ) else (  set vrednost /p = “Vnesi vrednost”  echo %vrednost%  ) |

6. Да се најде времето на извршување на секој од процесите P1, P2, P3, и P4 и просечното време на чекање по процес според табелата, ако процесите се извршуваат според FCFS, SJF with preemtion(со претпразнење) и RR алгоритмите. Циклусот за RR трае 4 временски единици.

|  |  |  |
| --- | --- | --- |
| Процес | Време на пристигнување | Време на извршување |
| Р1 | 0 | 5 |
| Р2 | 3 | 6 |
| Р3 | 5 | 8 |
| Р4 | 10 | 4 |

FCFS:

|  |  |  |  |
| --- | --- | --- | --- |
| P1 | P2 | P3 | P4 |

0 5 11 19 23

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Процес | Време на пристигнување | Време на извршување | **Време на постоење** | **Време на чекање** |
| Р1 | 0 | 5 | **5-0=5** | **5-5=0** |
| Р2 | 3 | 6 | **11-3=8** | **8-6=2** |
| Р3 | 5 | 8 | **19-5=14** | **14-8=6** |
| Р4 | 10 | 4 | **23-10=13** | **13-4=9** |

Времето на извршување(или постоење) на процесите е дадено во табелата.

Просечно време на чекање: (0 + 2 + 6 + 9) / 4 = 17/4 = 4.25

SJF with preemtion:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| P1 | P2 | P4 | P2 | P3 |

0 5 10 14 15 23

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Процес | Време на пристигнување | Време на извршување | **Време на постоење** | **Време на чекање** |
| Р1 | 0 | 5 | **5-0=5** | **5-5=0** |
| Р2 | 3 | 6 | **15-3=12** | **12-6=6** |
| Р3 | 5 | 8 | **23-5=18** | **18-8=10** |
| Р4 | 10 | 4 | **14-10=4** | **4-4=0** |

Времето на извршување(или постоење) на процесите е дадено во табелата.

Просечно време на чекање: (0 + 6 + 10 + 0) / 4 = 16/4 = 4.00

RR TQ = 4:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| P1 | P2 | P1 | P3 | P2 | P4 | P3 |

0 4 8 9 13 15 19 23

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Процес | Време на пристигнување | Време на извршување | **Време на постоење** | **Време на чекање** |
| Р1 | 0 | 5 | **9-0=9** | **9-5=4** |
| Р2 | 3 | 6 | **15-3=12** | **12-6=6** |
| Р3 | 5 | 8 | **23-5=18** | **18-8=10** |
| Р4 | 10 | 4 | **19-10=9** | **9-4=5** |

Времето на извршување(или постоење) на процесите е дадено во табелата.

Просечно време на чекање: (4 + 6 + 10 + 5) / 4 = 6.25

Системски софтвер група 2 14-12-2016

1. Што претставува контролен блок на процесот и што е овозможено со него?

Бидејќи процесот е динамичка структура и процесот е програма во состојба на извршување, оперативниот систем мора да знае каде да го продолжи извршувањето на дадениот процес откако ќе го стопира за да се ослободат ресурси за друг процес. Тоа се овозможува со контролниот блок. Контролниот блок е дел од меморијата односно мемориска структура со основни информации за процесот кој што OS го користи за управување со тој процес.

2. Како функционира редот за чекање на процесор?

Редот за чекање на процесор е линеарна листа која диспечерот го користи при управувањето со процесите. Редот за чекање на процесор е пополнет со процеси кои се во состојба ready.

Редот за чекање на процесор во суштина е поврзана листа формирана од контролните блокови на процесите. Има заглавие кое содржи информации за почетниот и последниот контролен блок во листата, и секој елемент во редот има покажувач кон следниот контролен блок. Самиот редослед по кои процесите чекаат во редот е дефиниран според некој алгоритам.

3. Што претставуваат поврзани листи кај алокација на меморијата? Објаснете!

Кај динамичката алогација меморијата се состои од процеси и празнини а оперативниот систем динамички води евиденција за зафатената меморија. Еден начин на водење е виденција за зафатената меморија е со помош на поврзани листи. Поврзаните листи се градат со записи, каде секој запис има одредени полиња како: **Тип на меморија**(дали е зафатена од процес), **Почетна адреса** за делот од меморијата кој ја опишува дадениот запис, **Должина на опишаната меморија**, и **покажувач кон следниот запис**.

4. Што претставува домен на заштита? Набројте неколку видови на напади на системот и објаснете три од нив!

Доменот на заштита е колекција од објекти и операции кои може да ги извршува даден процес ако е припадник на дадениот домен(во суштина, права). Секој домен дефинира множество на објекти и сите операции кои можат да се извршат на тоа множество.

Ако еден процес припаѓа на некој домен, тоа значи дека тој може да ги изврши операциите над објектите зададени во тој домен.

Неколку видови напади на оперативниот систем:

Лажирање со DNS, напаѓачот лажно се претставува како друг компјутер со IP адреси. Може да изврши напад на DNS сервисот давајќи му погрешни информации.

Smurf: Напаѓачот им праќа на сите членови на мрежата ICMP echo-request пакети и како одредишна адреса ја става адресата на жртвата а жртвата приме голем број на echo-reply пакети.

Смртоносен ping: Праќање на голем број на ICMP request пакети кои можат да бидат фатални за некои оперативни системи.

Нушкање, скенирање на приклучоци

5. Да се напише batch скрипта која во зависност од изборот на корисникот ќе ги извршува слениве акции:

1) копирање на сите фајлови од фолдер [c:\sistemski](file:///c:\sistemski) во [d:\sistemski](file:///d:\sistemski)

2) терминирање на процесите со PID 1224 и skype.exe

3) прикажување на датумот

|  |
| --- |
| @echo off  echo 1. Kopirane datoteki od [c:\sistemski](file:///c:\sistemski) do g:\sistemski  echo 2. Terminiranje na procesi so PID 1224 I ime skype.exe  echo 3. Prikazuvanje na datumot  set /p izbor="Izbor: "  if %izbor%==1 (  copy c:\sistemski g:\sistemski  ) else if %izbor% ==2 (  taskkill /pid 1224  taskkill /im skype.exe  ) else (  date /t  ) |

6. Да се најде времето на извршување на секој од процесите P1, P2, P3, и P4 и просечното време на чекање по процес според табелата, ако процесите се извршуваат според FCFS, SJF with preemtion(со претпразнење) и RR алгоритмите. Циклусот за RR трае 6 временски единици.

|  |  |  |
| --- | --- | --- |
| Процес | Време на пристигнување | Време на извршување |
| Р1 | 0 | 5 |
| Р2 | 4 | 6 |
| Р3 | 6 | 8 |
| Р4 | 9 | 4 |

FCFS:

|  |  |  |  |
| --- | --- | --- | --- |
| P1 | P2 | P3 | P4 |

0 5 11 19 23

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Процес | Време на пристигнување | Време на извршување | **Време на постоење** | **Време на чекање** |
| Р1 | 0 | 5 | **5-0=5** | **5-5=0** |
| Р2 | 4 | 6 | **11-4=7** | **7-6=1** |
| Р3 | 6 | 8 | **19-6=13** | **13-8=5** |
| Р4 | 9 | 4 | **20-9=11** | **11-4=7** |

Времето на извршување(или постоење) на процесите е дадено во табелата.

Просечно време на чекање: (0 + 1 + 5+ 7)/4 = 13/4 = 3.25

SJF with preemtion:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| P1 | P2 | P4 | P2 | P3 |

0 5 9 13 15 23

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Процес | Време на пристигнување | Време на извршување | **Време на постоење** | **Време на чекање** |
| Р1 | 0 | 5 | **5-0=5** | **5-5=0** |
| Р2 | 4 | 6 | **15-4=11** | **11-6=5** |
| Р3 | 6 | 8 | **23-6=17** | **17-8=9** |
| Р4 | 9 | 4 | **13-9=4** | **4-4=0** |

Времето на извршување(или постоење) на процесите е дадено во табелата.

Просечно време на чекање: (5 + 9) / 4 = 14/4 = 3.5

RR TQ = 6:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| P1 | P2 | P3 | P4 | P3 |

0 5 11 17 21 23

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Процес | Време на пристигнување | Време на извршување | **Време на постоење** | **Време на чекање** |
| Р1 | 0 | 5 | **5-0=5** | **5-5=0** |
| Р2 | 4 | 6 | **11-4=7** | **7-6=1** |
| Р3 | 6 | 8 | **23-6=17** | **17-8=9** |
| Р4 | 9 | 4 | **21-9=12** | **12-4=8** |

Времето на извршување(или постоење) на процесите е дадено во табелата.

Просечно време на чекање: (0 + 1 + 9 + 8) / 4 = 18/4 = 4.5

Системски софтвер група 3 07-02-2018

1. Објаснете го проширениот дијаграм на состојби на процесите.

Основниот дијаграм на состојби на процесите се состои од 5 состојби и може да се прикаже со конечен автомат. 5-те состојби се: Start, Stop, Ready, Wait, Run.

Во реалните оперативни системи постои можност процесот да се прекине со работа привремено. Во таквите оперативни системи тоа може да го направи корисник човек или самиот оперативен систем. Таквото прекинување на процесот се нарекува суспендирање што значи дека треба да се воведи проширен дијаграм на состојби.

Во проширениот дијаграм на состојби се воведуваат две нови состојби: Suspend Ready и Suspend Wait од причина што суспендирање на процесот може да се направи само кога процесот е во состојба Ready или состојба Wait.

Новите транзиции се:

Wait → Suspend Wait се доведува процесот од состојба Wait во состојба Suspend Wait

Suspend Wait → Suspend Ready ресурсот неопходен за понатамошно извршување на процесот е слободен, но процесот е и понатаму суспендиран

Suspend Wait → Wait процесот се ослободува од суспендирање но ресурсот кој е неопходен за процесот не е слободен. Оваа транзиција е можна на експлицитно барање на корисникот човек.

Suspend Wait → Wait процесот се ослободува од суспендирање и се става во редот за чекање процес. Оваа транзиција е можна на експлицитно барање на корисникот човек.

2. Што претставува диспечер на системот? Што е каснење на диспечерот?

Диспечерот е дел од програмата или програмите за распоредување на процесите за извршување.

Задачата на диспечерот е да доделува процесор на процесите кои се во состојба Ready и се наоѓаат во процесорскиот ред. За разлика од планерот на работи, диспечерот работи со процесите кои се во редот на чекање за процесор додека планерот се повикува кога ќе се појават нови процеси или завршат постоечки.

Времето за кое диспечерот застанува процес и му доделува процесор на другиот, е познато како каснење на диспечерот. Практично времето на одзив на диспечерот е исто така клучно за перформансите на оперативниот систем, што помало е ова време, тоа подобро.

3. Што е разликара помеѓу мрежен и дистрибуиран оперативен систем?

Кај мрежните оперативни системи, автономијата е голема зашто во мрежа се вклучуваат индивидуални компјутери, секој со свој оперативен систем. Единствено заеднички карактеристики се протоколите со кој комуницираат во мрежата.

Кај дистрибуираните оперативни системи, оперативниот систем е еден и единствен кој управува со сите ресурси во мрежата. Корисникот човек може да го гледа системот како еднопроцесорски. Ова значи дека покрај тоа што се овозможува споделување на датотеки, овозможува и споделување на процесите. Оперативниот систем тука е наменет за работа со повеќе процесори кои се флексибилно поврзани во мрежа.

4. Да се анализира состојбата на меморијата ако се користи континуирано доделување на меморија и партиции со променлива големина за динамичко доделување на меморија според систем на здружени парови! Целата меморија е 2MB, а процесите А = 110 КB, B=35KB. Објаснете!

За управување на меморијата во системот на здружени парови се користи по една листа за секоја од големините на меморијата како 20, 21, се до вкупната големина на меморијата(од примерот до 211), На почеток, не постојат партиции со такви големини, па сите листи се празни, освен онаа листа за партицијата од 2MB. Кога се доделува процес во меморијата, партицијата се дели на делови се додека не наидеме на најмалата партиција поголема од меморијата потребна за процесот.

Пред додавање процес:

-H(2MB)

По додавање на процесот А

-H(1MB)

-H(512KB)

-H(256KB)

-H(128KB)

-P(128KB од кои: 110KB искористени, 18KB слободни)

По додавање на процесот B:

-H(1MB)

-H(512KB)

-H(256KB)

-H(64KB)

-P(64KB од кои: 35KB искористени, 29KB слободни)

Заклучот: Се јавува интерна фрагментација.

5. Што претставува домен на заштита? Набројте неколку видови на напади на системот и објаснете три од нив!

Доменот на заштита е колекција од објекти и операции кои може да ги извршува даден процес ако е припадник на дадениот домен(во суштина, права). Секој домен дефинира множество на објекти и сите операции кои можат да се извршат на тоа множество.

Ако еден процес припаѓа на некој домен, тоа значи дека тој може да ги изврши операциите над објектите зададени во тој домен.

Неколку видови напади на оперативниот систем:

Лажирање со DNS, напаѓачот лажно се претставува како друг компјутер со IP адреси. Може да изврши напад на DNS сервисот давајќи му погрешни информации.

Smurf: Напаѓачот им праќа на сите членови на мрежата ICMP echo-request пакети и како одредишна адреса ја става адресата на жртвата а жртвата приме голем број на echo-reply пакети.

Смртоносен ping: Праќање на голем број на ICMP request пакети кои можат да бидат фатални за некои оперативни системи.

Нушкање, скенирање на приклучоци

6. Што е DMA? Како се зголемува конкурентноста на системот со DMA? Објаснете!

Еден начин на комуникација со влезно-излезните уреди за меморија е програмиран. Тоа значи дека инструкциите кои процесорот ги извршува иницираат комуникација со даден в-и уред. Недостаток на овој начин е тоа што се намалува конкурентноста затоа што процесорот мора да чека за да се пренесат податоците(особено ако е големо количество податоци).

DMA(Direct Access Memory) има токму за цел овој недостаток да се избегни, а со тоа и да се зголеми конкурентноста на системот.

Работи на тој начин што Секој бајт поминува низ DMA контролерот, а потоа до главната меморија, или до в-и уред. Во DMA контролерот има соодветни бројачи за преостанати бајти и следна адреса до бајтот кој ќе се пренесува.

На крај, DMAгенерира прекинувачки сигнал до процесорот со цел да сигнализира комплетирање на преносот на податоци.

7. Да се напише batch скрипта која во зависност од изборот на корисникот ќе ги извршува слениве акции:

а) терминирање на процесот со име skype.exe

б) печатење – looping низа на броеви од 50 до 25 со чекор -1

в) ќе се изврши терминирање на процесот со PID 7576

|  |
| --- |
| @echo off  echo 1. Terminiranje na procesot so ime skype.exe  echo 2. Pecatenje so loop  echo 3. Terminiranje na procesot so PID 7576  set /p izbor="Izbor: "  if %izbor%==1 (  taskkill /im skype.exe  ) else if %izbor% ==2 (  for /L %%n in (50,-1,25) do (  echo %%n )  ) else (  taskkill /pid 7576  ) |

8. Да се најде времето на извршување на секој од процесите P1, P2, P3, и P4 и просечното време на чекање по процес според табелата, ако процесите се извршуваат според FCFS, SJF with preemtion(со претпразнење) и RR алгоритмите. Циклусот за RR трае 3 временски единици. Според кој од алгоритмите процесот P3 ќе заврши најбрзо.

|  |  |  |
| --- | --- | --- |
| Процес | Време на пристигнување | Време на извршување |
| Р1 | 0 | 7 |
| Р2 | 4 | 6 |
| Р3 | 6 | 8 |
| Р4 | 9 | 5 |

FCFS:

|  |  |  |  |
| --- | --- | --- | --- |
| P1 | P2 | P3 | P4 |

0 7 13 21 26

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Процес | Време на пристигнување | Време на извршување | **Време на постоење** | **Време на чекање** |
| Р1 | 0 | 7 | **7-0=7** | **7-7=0** |
| Р2 | 4 | 6 | **13-4=9** | **9-6=3** |
| Р3 | 6 | 8 | **21-6=15** | **15-8=7** |
| Р4 | 9 | 5 | **26-9=17** | **15-5=10** |

Времето на извршување(или постоење) на процесите е дадено во табелата.

Просечно време на чекање: (0 + 3 + 7 + 10)/4 = 20/4 = 5.00

SFJ with preemtion:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| P1 | P2 | P4 | P2 | P1 | P3 |

0 4 9 14 15 18 26

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Процес | Време на пристигнување | Време на извршување | **Време на постоење** | **Време на чекање** |
| Р1 | 0 | 7 | **18-0=18** | **18-7=11** |
| Р2 | 4 | 6 | **15-4=11** | **11-6=5** |
| Р3 | 6 | 8 | **26-6=20** | **20-8=12** |
| Р4 | 9 | 5 | **14-9=5** | **5-5=0** |

Времето на извршување(или постоење) на процесите е дадено во табелата.

Просечно време на чекање: (11 + 5 + 12 + 0)/4 = 28/4= 7.00

RR TQ=3:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| P1 | P1 | P2 | P3 | P1 | P4 | P2 | P3 | P4 | P3 |

0 3 6 9 12 15 16 19 22 24 26

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Процес | Време на пристигнување | Време на извршување | **Време на постоење** | **Време на чекање** |
| Р1 | 0 | 7 | **15-0=15** | **15-7=8** |
| Р2 | 4 | 6 | **19-4=15** | **15-6=9** |
| Р3 | 6 | 8 | **26-6=20** | **20-8=12** |
| Р4 | 9 | 5 | **24-9=15** | **15-5=10** |

Времето на извршување(или постоење) на процесите е дадено во табелата.

Просечно време на чекање: (8 + 9 + 12 + 10)/4 = 39/4= 9.75

Според гантограмите, процесот P3 завршува најбрзо во FCFS алгоритмот(во временската единица 21!)

Според вкупното време на постоење P3 постоел најмалку во FSFS(15 временски единици!)

Системски софтвер група 4 30-01-2017

1. Како функционира редот за чекање на процесор?

Редот за чекање на процесор е линеарна листа која диспечерот го користи при управувањето со процесите. Редот за чекање на процесор е пополнет со процеси кои се во состојба ready.

Редот за чекање на процесор во суштина е поврзана листа формирана од контролните блокови на процесите. Има заглавие кое содржи информации за почетниот и последниот контролен блок во листата, и секој елемент во редот има покажувач кон следниот контролен блок. Самиот редослед по кои процесите чекаат во редот е дефиниран според некој алгоритам.

2. Што претставува процес? Што вклучува процесот?

Процесот е програма во состојба на извршување и покрај програмскиот код и двете фундаментални мемориски секции(stack секција и секција на податоци со глобални променливи) тој вклучува:

* вредност на програмскиот бројач
* вредност на останатитите важни регистри на процесорот
* влезно-излезни ресурси кои што процесор ги користи
* контролен блок на процесот
* опишувач на процесор(process descriptor)

3. На кои три основни целини може да се подели јадрото според теоретскиот модел на ОС?

* Прво ниво за обработка на прекини
  + одговара на надворешни прекини и системски повици
* Диспечер на системот
  + дел од јадрото кој го доделува процесот на процесорите
* Рутини за остварување на интерпроцесни комуникации
  + обезбедува комуникација помеѓу процесите

4. Од што се состои типичен влезно-излезен приклучот? Објаснете!

Контролерот може да работи како:

* приклучок(port)
* магистрала(bus)
* уред(device)

Ако работи како приклучок, тогаш тој се состои од четири вида на регистри:

* Контролен регистер кој служи за нагодување на комуникацијата и режимот на работа на уредот. Практично, процесорот нагодува како ќе управува со уредот па затоа процесорот исклучиво запишува во овој регистер
* Статусен регистер кој самиот уред го полни. Има за цел да информира за состојбата на командите(дали се извршени или не), сигнализација за крај на командата или спремност. Што значи дека тука процесорот исклучиво чита од овој регистер.
* Регистер на податоци за влезен режим, може да игра улога на бафер за влезно-излезниот уред од причина што податоците што процесорот треба да ги прочита од уредот се сместуваат тука.
* Регистер на податоци за излезен режим, исто така може да игра улога на бафер бидејќи во овој регистер процесорот ги запишува податоците што треба да одат кон уредот.

5. Што претставува домен на заштита? Набројте неколку видови на напади на системот и објаснете два!

Доменот на заштита е колекција од објекти и операции кои може да ги извршува даден процес ако е припадник на дадениот домен(во суштина, права). Секој домен дефинира множество на објекти и сите операции кои можат да се извршат на тоа множество.

Ако еден процес припаѓа на некој домен, тоа значи дека тој може да ги изврши операциите над објектите зададени во тој домен.

Неколку видови напади на оперативниот систем:

Лажирање со DNS, напаѓачот лажно се претставува како друг компјутер со IP адреси. Може да изврши напад на DNS сервисот давајќи му погрешни информации.

Smurf: Напаѓачот им праќа на сите членови на мрежата ICMP echo-request пакети и како одредишна адреса ја става адресата на жртвата а жртвата приме голем број на echo-reply пакети.

Нушкање, скенирање на приклучоци, смртоносен ping

6. Во кои случаи може да се користи техниката на преклопување(overlay technique)? Како таа функционира?

Техниката на преклупување е дел од мноштвото на програмерски техники за управување со меморијата при извршување на програмите.

Се користи во ситуации кога потребната меморија за извршување на дадена програма е поголема од физичката меморија на компјутерскиот систем.

Техниката на преклопување функционира на тој начин што програмерот ја организира програмата на соодветни компоненти(пример два независни делови кои се извршуваат едно по друго, заеднички рутини, и заеднички податоци. Ако ја земиме организацијата на програмата како во примеров, ќе има два overlays. Едниот overlay ќе го содржи првиот дел кој треба да се извршува, заедничките рутини и податоци, а другиот overlay ќе го содржи вториот дел кој треба да се извршува заедно со заедничките рутини и податоци. И кај двата overlays мора да има overlay driver кој управува со дадените overlays. Кога ќе се преклопи едниот overlay со следниот, се зафаќа голем дел од истата алоцирана меморија.

7. Да се напише batch скрипта која во зависност од изборот на корисникот ќе ги извршува слениве акции:

1) прикажување на датумот

2) копирање на сите фајлови од фолдер [c:\sistemski](file:///c:\sistemski) vo d:\sistemski

3) проверка дали постои фолдерот [D:\sistemski](file:///D:\sistemski) и доколку не постои да се креира

|  |
| --- |
| @echo off  echo 1. Prikazuvanje na datumot  echo 2. Kopirane datoteki od c:\sistemski do g:\sistemski  echo 3. Proverka dali postoi D:\sistemski  set /p izbor="Izbor: "  if %izbor%==1 (  date /t  ) else if %izbor% ==2 (  copy c:\sistemski g:\sistemski  ) else (  if not exist [d:\sistemski](file:///d:\sistemski) (  mkdir [d:\sistemski](file:///d:\sistemski)  )  ) |

8. Да се најде времето на извршување на секој од процесите P1, P2, P3, и P4 и просечното време на извршување по процес според табелата, ако процесите се извршуваат според FCFS, SJF with preemtion(со претпразнење) и RR алгоритмите. Циклусот за RR трае 4 временски единици. Според кој од алгоритмите процесот P3 ќе заврши најбрзо.

|  |  |  |
| --- | --- | --- |
| Процес | Време на пристигнување | Време на извршување |
| Р1 | 0 | 8 |
| Р2 | 1 | 4 |
| Р3 | 2 | 9 |
| Р4 | 5 | 5 |

FCFS:

|  |  |  |  |
| --- | --- | --- | --- |
| P1 | P2 | P3 | P4 |

0 8 12 21 26

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Процес | Време на пристигнување | Време на извршување | **Време на постоење** | **Време на чекање** |
| Р1 | 0 | 8 | **8-0=8** | **8-8=0** |
| Р2 | 1 | 4 | **12-1=11** | **11-4=7** |
| Р3 | 2 | 9 | **21-2=19** | **19-9=10** |
| Р4 | 5 | 5 | **26-5=21** | **21-5=16** |

Времето на извршување(или постоење) на процесите е дадено во табелата.

Просечно време на извршување: (8 + 11 + 19 + 21)/4 = 59/4 = 14.75

SJF with preemtion:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| P1 | P2 | P4 | P1 | P3 |

0 1 5 10 17 26

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Процес | Време на пристигнување | Време на извршување | **Време на постоење** | **Време на чекање** |
| Р1 | 0 | 8 | **17-0=17** | **17-8=9** |
| Р2 | 1 | 4 | **5-1=4** | **4-4=0** |
| Р3 | 2 | 9 | **26-2=24** | **24-9=15** |
| Р4 | 5 | 5 | **10-5=5** | **5-5=0** |

Времето на извршување(или постоење) на процесите е дадено во табелата.

Просечно време на извршување: (17+4+24+5)/4 = 47/4 = 11.75

RR TQ = 4:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| P1 | P2 | P3 | P1 | P4 | P3 | P4 | P3 |

0 4 8 12 16 20 24 25 26

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Процес | Време на пристигнување | Време на извршување | **Време на постоење** | **Време на чекање** |
| Р1 | 0 | 8 | **16-0=16** | **16-8=8** |
| Р2 | 1 | 4 | **8-1=7** | **7-4=4** |
| Р3 | 2 | 9 | **26-21=24** | **24-9=15** |
| Р4 | 5 | 5 | **25-5=20** | **20-5=15** |

Времето на извршување(или постоење) на процесите е дадено во табелата.

Просечно време на извршување: (16 + 7 + 24 + 20)/4 = 67/4 = 11.75

Според гантограмите, процесот P3 завршува најбрзо во FCFS алгоритмот(во временската единица 21!)

Според вкупното време на постоење P3 постоел најмалку во FSFS(19 временски единици!)

1.Како функционира редот за чекање на процесор? Test 1 Сите процеси кои се спремни за работа се наоѓаат во работната меморија и се чуваат за редот на чекање на процесор. Постојат два вида на редови на чекање : -Ред на чекање на процесор, -Ред на чекање на влезно-излезен уред. -Редовите на чекање на процесор по правило се реализираат како поврзани листи формирани од контролни блокови на процеси со дефиниран редослед на извршување на процесорот. Редоследот се задава преку заглавие на листата кое што содржи информација за почетниот и последниот контролен блок во листата и покажувач на следниот контролен блок. Оперативниот систем води посебен ред на чекање за секој влезно-излезен уред. Секој ред на чекање уред содржи поврзани листи на контролни блокови на процеси кој што тој уред ги бара. 2.Што претствавува процес? Што вклучува процесот ? -Процесот е програма или дел од програмата во состојба на извршување заедно со сите ресурси кои се потребни за работата на програмата. -Самиот оперативен систем е составен од низа процеси. -Секој процес има 3 секции: -Програмска или текстуална секција која не се менува и која содржи програмски код; -Стек секција која содржи привремени податоци; -Секција на податоци која содржи глобални променливи. -Процесот опфаќа и: -Вредност на програмскиот бројач (program counter); -Вредност на останатите важни регистри на процесорот, како и -Влезно-излезни ресурси. 3.На кои три основни целини може да се подели јадрото според теоретскиот модел на ОС? -Прво ниво за обработка на прекини (First Level Interrupt Handler - FLIH), -Првото ниво го сочинуваат рутини за одредување на изворот за прекин. После извршувањето на кодот прекинот се смета за остварен, а диспечерот одлучува на кој процес ќе го даде процесорот за користење. -Диспечер на системот -Диспечерски дел од јадрото кој го доделува процесот на процесорите, а при тоа процесорот секогаш се доделува врз основа на некој алгоритам како што е Shortest Job First, Рутини за остварување на интерпроцесни комуникации Овие рутини овозможуваат комуникација помеѓу процесите 4.Од што се состои типичен влезно-излезен приклучок (објаснете!) -Се состои од 4 вида на регистри: -Контролен регистер служи за поставување режим на работата на уредот како што се брзината на приклучокот и типот на комуникација. -Статусен регистер го опишува статусот на командите што ги извршува, односно дали командата е извршена, дали е спремен податокот, дали настанала грешка и.т.н. -Влезен Регистер на податоци ( Data-in) служи за читање на податоци од влезните уреди. -Излезен Регистер на податоци (Data-out) служи за запишување на податоците на излезниот уред. 5.Што претставува домен на заштита? Набројте неколку видови на напади на системот и објаснете два! -ОС управува со различни хардверски и софтверски објекти. Секој објект има име и кон него може да се пристапи преку прецизно дефинирана операција. Проблемот на заштита се сведува на контрола на пристапот кон објектите на ОС : на објектите може да им пристапат само оние корисници кои се овластени и над објектот можат да се извршат само дозволени операции. Секој домен дефинира множество на објекти и сите операции кои можат да се извршат на тоа множество. Можноста да се изврши операција над објектот се нарекува право на пристап (access right). Доменот претставува колекција од права на пристап на дефинирани парови. -Системот може да биде нападнат на повеќе начини : Одбивање на услуга(DoS) , лажирање ( spoofing) , нушкање ( sniffing) и други. ЛАЖИРАЊЕ (spoofing) – При овој начин на напад напаѓачот ги следи ИП адресите и се прествавува како друг компјутер. Бидејќи DNS не го проверува изворот на информацијата односно од каде таа доаѓа , напаѓачот може да изврши напад на DNS сервисот давајќи му погрешни информации. Најдобра заштита од овој вид на напад е спречување на рутирањето на датаграмите со неисправни изворни адреси. -НУШКАЊЕ (sniffing) – Метод во кој со специјална програма наречена нушкач (Sniffer) се пресретнуваат TCP/IP пакети кои поминуваат низ одреден компјутер и по потреба се прегледува нивната содржина. Доколку во мрежата се забележат нешифрирани податоци , sniffer-от лесно може да дојде до доверливите информации. 6.Во кои случаи може да се користи техниката на преклопување ( overlay technique?) Како таа функционира? -За да се овозможи извршување на процесот кој е поголем и од самата физичка меморија се користи техниката на преклопување. Преклопувањето овозможува во меморијата да се чуваат само оние делови на програмата кои се потребни во тој момент. Кога ќе бидат потребни другите делови тие ќе се вчитаат во меморијата на местото на деловите кои веќе не се потребни. -Техниката на преклопување ја забавува работата на самата програма, бидејќи деловите на програмата се вчитуваат во повеќе интерации, но со неа се постигнува нешто што поинаку не може да се постигне – извршување на програма која е поголема од оперативната меморија. Таа може да биде корисна на системите со ограничени ресурси. 1.Кои се состојбите во кои може да се најде процесот ? test2 -Процесот може да се најде во неколку состојби (5 или 7 зависно од конечниот автомат), а 3 се најважни -Состојба на извршување (RUN,RUNNING) процесорот извршува инструкции на овој процес. -Состојба на чекање на процесор (READY, RUNNABLE) процесот ги добил сите потребни ресурси освен процесорот. -Состојба на чекање на ресурс (WAIT, UNRUNNABLE) процесот чека на некое случување бидејчи за извршување на процесот е потребен ресурс. 2.Објаснете ја основната функција на јадрото на ОС? Определување на процес т.е. создавање на опркужување во кое може да постојат процеси на процесорите и обезбедување на механизми за интерпроцесна комуникација; Како на еден процесор во еден момент може да се изврши еден процес; Јадрото одредува кога и во кое време процесот ќе добие процесор. Оваа појава е позната како мултиплексирање. За да јадрото ја реализира својата ф-ја неопходно е да поседува одредени компоненти: -Механизам за прекинување –обезбедува извршување на управувачката програма -Заштитен механизам за адресирање на меморија – спречува погрешно адресирање -Збир на прилевигирани инструкции. -Часовник за реално време (real-time clock) 3.Што претставува интерна фрагментација? -Два процеси никако не смеат да бидат сместени во една партиција. Двата вида на ред за чекање се карактеризираат со интерна фрагментација – деловите на меморијата кои се поголеми од процесот се потполно неискористени. -Интерната фрагментација се јавува доколку на процесот му се додели мемориска партиција поголема од меморијата која ја бара процесот. Останатата меморија е неупотреблива. 4.Кои се основните разлики меѓу корисничките и јадрените нишки? -За разлика од корисничките нитки, нитките на јадрото не мора да припаѓаат на процес; -Распоредувањето на процесорот за корисничките нитки го извршува библиотека за работа со корисничките нитки, додека распоредувањето за нитките на јадрото го извршува процесорот; -Јадрото не е свесно за корисничките нитки 5.Што е DMA? Како се зголемува конкурентноста на системот со DMA? -DMA(Direct Memory Access)се користи за да се избегне програмирањето на ВИ уреди за трансфер на големи количини на податоци. Техника на програмиран ВИ – кога уредот ги подготви податоците, процесорот треба да го преземи секој бајт од контролерот. При тоа, процесорот секој пат го проверува статусниот бит, што значи се извршуваат два ВИ циклуци. -DMA контролер – за да може да се направи брз пренос на податоци кај уредите како што е дискот на релација контролер – работна меморија и обратно -Со DMA се бајпасира процесорот при трансферот меѓу ВИ уред и меморијата 6.Кои се програмските и системските закани по сигурноста на системот? Објаснете две! -Тројански коњи, Замка (trap door), Преполнување на стекот или баферот. -Вируси, Компјутерски црви (worms) -Морисовцрв. -Тројански коњи се поддметнати во кодот на програмата со цел да се промени ф-јата на оригиналната програма. Тоа се програми кои се преправаат како да се корисни алатки, но во реалноста се злонамерни и многу опасни . Тројанците се способни да ја намалат сигурноста на нашиот компјутер, да дозволат на другите целосен пристап до сите датотеки на компјутерот. Вообичаен тројански коњ се состои од две програми , Client и Server. -Client – Овој дел го извршува хакерот. Тој гледа екран со неколку видови команди. Типичен пример за ваков вид програма е Sub7. Server – Серверот е програма која се наоѓа на нашиот компјутер и ние мораме да ја стартуваме за хакерот да има било каква можност да ја користи. Овие програми се мали и кога еднаш че се извршат се сместуваат сами себеси во разни директориуми и се скриени. Со секое стартување на компјутерот се стартуваат и тие. -Морисов црв – Морисовиот црв во 1988 година направи голема штета на UNIX системите. Овој црв се состои од главна програма која се обидува да ги открие лозинките на корисникот и јадица. Заразениот систем со техниката на преполнување на баферот или стекот се обидува да вметне јадица на друг систем во мрежата кориистејќи ги слабостите на некои програми. Кога црвот ќе гозарази другиот компјутер ,понатаму двете машини продолжуваат да го шират црвот низ мрежата. TEST3>> 1.Што претставува процес? Што претставува контролен блок на процесот и што е овозможено со него ? -Процесот е динамичка структура и најлесно може да се опише како програма која е во состојба да се извршува како низа активности кои се последица од извршувањето на програмата.Дел од процесот се и податоци кои се неопходни за негово управување. Овие податоци ги генерира и ги користи ОС, односно диспечерот , а се наречени Контролен блок на процесот и дескриптор на процесот. -Контролниот блок е дел од меморијата односно мемориска структура со основни информации за процесот кој што ОС го користи за управување со тој процес. Со него програмата може да се прекинува и повторува повеќе пати.Контролниот блок се прикажува до процесот, но во реалниот ОС никогаш не се наоѓа во исто меморско подрачје. 2.Што се регистрите, која е нивната функција и какви видови постојат? -Регистер претставува хардверски уред за чување на бинарни податоци. Регистрите служат за привремено сместување на податоци кои се обработуваат или се резултат на пресметки -AC, MAR, MBR, PC, IR, InREG, OutReg, Status и други регистри. 3.Која е задачата и како функционира диспечерот? -Задача на диспечерот е да доделува процесор на процесите кои што се наоѓаат во процесорскиот ред. Диспечерот потврдува на кој процес е најдобро да се додели процесорот. Диспечерот прво одредува дали процесорот треба да му се додели на тековниот процес или на некој друг процес . Може да се повикува многу често . Времето за кое диспечерот застанува процес и му доделува процесор на другиот е познато како каснење на диспечерот. 4.Како се реализира индиректна комуникација меѓу процесите! Објаснете една од постапките ? -Пораките се испраќаат и примаат преку: Порт и поштенско сандаче -Поштенско сандаче е објект во кој ОС ги остава пораките од другите процеси. -Врската се воспоставува меѓу парови процеси кои го делат сандачето. Кон врската може да се придружат повеќе од два процеси. Меѓу секој пар процеси може да постојат повеќе различни врски каде секоја одговара на едно сандаче. ОС мора да обезбеди механизам за правење на ново сандаче за секој пар процеси кои сакаат да комуницираат. -ПРИМЕР: три процеси делат исто сандаче. Првиот процес испраќа порака во А а вториот и третиот проверуваат дали има порака во А. Кој процес ќе ја земе пораката? -Решение: -Со дозвола да се воспостави врска преку поштенско сандаче меѓу два процеси, -Со забрана за истовремено примање на пораки од поголем број процеси, - Со адресирање на пораката. 5.Што претставува спулер? Објаснете! -Спулер е бафер кој привремено ги чува излезните податоци наменети за некој неделлив уред и овозможува пристап до тие уреди на следниов начин: -Процесите ги запишуваат податоците наменети за уредот на диск, а оперативниот систем управува со спулерот на начин што ги опслужува барањата едно по едно. -Предноста во користењето на спулерот е тоа што процесорот релативно брзо го поставува своето барање во бафер, а после тоа слободно понатаму ги извршува своите активности 6.Објаснете ги следните алгоритми за доделување на меморијата: a) first-fit , b) best-fit , c) worst-fit , d) next-fit, e) Како може да се оправда користењето на worst-fit алгоритмот за доделување на меморија ? Што е неговата потенцијална предност во однос на best-fit ? -First-fit – На процесот му се доделува првата празнина која е доволно голема. Пребарувањето се одвива или од почетокот на листата на празнини или од местото до каде сме застанале со претходното пребарување. Времето на пребарување е минимално. Се доделува првата празнина на која ќе се наиде а е доволно голема без оглед дека понатаму може да има и друго која е поповолна. -Best-fit -На процесот му се доделува најмалата празнина која е доволно голема за сместување на процесот. Се пребарува целата листа. Ако листата е подредена по големина не мора да се пребарува цела. -Worst-fit – На процесот му се доделува најголемата празнина. Се пребарува цела листа. Првите два се подобри по перформанси и по искористување на меморјата. -Next-fit – Работи полоши од first fit. Ја пребарува моморијата од локацијата на последно алоцираниот блок и го избира следниот слободен блок кој е доволно голем. Многу често се алоцира мемориски блок на крајот на меморијата за да биде најголем. -Сместува во најголемиот неискористен блок. Истото од под worst-fit) 7.Што претставува домен на заштита? На кои начини може да се изврши идентификација на корисникот? Набројте неколку видови на напади на системот и објаснете два ! -ОС управува со различни хардверски и софтверски објекти. Секој објект има име и кон него може да се пристапи преку прецизно дефинирана операција. Проблемот на заштита се сведува на контрола на пристапот кон објектите на ОС : на објектите може да им пристапат само оние корисници кои се овластени и над објектот можат да се извршат само дозволени операции. Секој домен дефинира множество на објекти и сите операции кои можат да се извршат на тоа множество. Можноста да се изврши операција над објектот се нарекува право на пристап (access right). Доменот претставува колекција од права на пристап на дефинирани парови. -Идентификација може да се оствари на 3 начини: -Специјален хардвер како што е клуч или ID картичка; -внесување на доверливи информации, како што е лозинката; -биолошките атрибути на корисникот;

-Системот може да биде нападнат на повеќе начини:

Одбивање на услуга(DoS), лажирање ( spoofing), нушкање ( sniffing) и други.

ЛАЖИРАЊЕ (spoofing) – При овој начин на напад напаѓачот ги следи ИП адресите и се прествавува како друг компјутер. Бидејќи DNS не го проверува изворот на информацијата односно од каде таа доаѓа , напаѓачот може да изврши напад на DNS сервисот давајќи му погрешни информации. Најдобра заштита од овој вид на напад е спречување на рутирањето на датаграмите со неисправни изворни адреси.

-НУШКАЊЕ (sniffing) – Метод во кој со специјална програма наречена нушкач( Sniffer) се пресретнуваат TCP/IP пакети кои поминуваат низ одреден компјутер и по потреба се прегледува нивната содржина. Доколку во мрежата се забележат нешифрирани податоци , sniffer-от лесно може да дојде до доверливите информации.

1.Што вклучува процесот ? Објаснете го проширениот дијаграм на состојби на процесот? Test4 Секој процес има 3 секции : -Програмска или текстуална секција која не се менува и која содржи програмски код; -Стек секција која содржи привремени податоци; -Секција на податоци која содржи глобални променливи. Процесот опфаќа и : Вредност на програмскиот бројач (program counter); -Вредност на останатите важни регистри на процесорот ,Влезно-излезни ресурси кои што ги користи како и разни други датотеки и влезно излезни уреди

2.Под кои услови се појавува застој? Дали е можно системот да се најде во состојба на застој доколку на него постои само еден процес ? Објаснете! -Застој настанува ако истовремено се исполнуваваат следниве услови. -При користење на ресурсите се почитува принципот на меѓусебно исклучување; -Доделувањето на ресурсите се обавува без претпразнење-на процесот не може да му се одземе ресурс; -Процесот задржува еден ресурс бидејќи му е потребен и чека за ресурс кој го користи некој друг процес; -Кружно чекање, -Ако на системот постои само еден процес не е можно да се појави застој. Ова е директна последица на условите на задржување на ресурсите и чекање на други ресурси.

3.Кои хардверски предуслови се потребни за реализација на функциите на јадрото на оперативниот систем, односно за надградба на хардверот со јадро во хиерархиски систем (кои компоненти треба да ги поседува хардверот ?) Компонентите кој хардверот треба да ги исполни спаѓаат: Механизам за прекинување обезбедува извршување на управувачката програма, Заштитен механизам за адресирање на меморија – спречува погрешно адресирање, -Збир на привилегирани инструкции, и – часовник на реално време.

4.Што претставуваат поврзани листи кај алокација на меморијата? Објаснете! -Алокација на меморија е динамичка –меморијата се состои од процеси и празнини, оперативниот систем динамички води евиденција за зафатената меморија. -Поврзаи листи се градат од записи со следниве структури: -Прво поле на записот означува тип на меморија. –Второ поле ја содржи адреса на делот на меморијата кој ја опишува дадениот запис. – Трето поле ја означува должината на опишаната меморија. –Четврто поле содржи покажувач кон следниот запис. Управувањето се одвива: -При заземање на меморијата, во поврзана листа се бара празнина(слободна мем) со дозволена големина. – Ако најде соодветна празнина, се започнува процес, а евинтуално се пушта и нов празен јазол. – кога процесот А ќе заврши со работа се ослободува зафатената меморија.

5.Во што е разликата помеѓу распоредување на процеси кај процесорите без пред распоредување со предпразнење? -Распоредување со претпразнење: Процесорот може да се одземе од процес кој не завршил со своите активности, неопходна хардверска компонента за реализација е тајмер кој поставува прекинувачки сигнал, претпразнењето влијае врз конструкцијата на јадрото -Распоредување без претпразнење: Процесот може да се одземе само од процес кој ги завршил своите активности или чека некој ресурс.

6.Набројте ги основните функции на влезно-излезниот потсистем! Објаснете ги а незавнисност на уредите! Која е последицата од незавнисноста на уредите? -Ги управува влезно-излезните уреди и операциите кои тие уреди ги извршуваат и ги контролираат; -Обезбедува што поедноставен инрерфејс кон корисникот и остатокот од системот. Постојат два аспекти на независност на уредите : -Независноста на програмата од конкретниот модел на уредот кој се користи. -Независност на програмата од конкретниот вид на уредот кој се користи. Последица : Програмите не работат со реални конкретни уреди туку сите ВИ операции ги обавуваат со помош на виртуелни уреди. Примери на виртуелни уреди се ТЕКОВИ и ДАТОТЕКИ

6.Објаснете ги следниве напади: a) DoS, b) лажирање (spoofing) и c) Нушкање ( snoofing). А) DoSПредизвикува стопирање на работата на сервисот или програмата така што другите корисници им се оневозможува да работат со нив. Овој напад може да се изврши на мрежниот слој преку праќање на голем број на SYN пакети TCP. Заштита може да се воспостави преку контролирање на бтојот на SYN пакети во единица време.

1.Што претставуваат мрежни системи/дистрибуирани? (Споредба)

Мрежно оперативните системи ги карактеризираат компјутерите во мрежа. Овие компјутери содржат **релативно висок степен на автономија - секој компјутер има свој оперативен систем**, и се во можност меѓусебно да разменуваат податоци со помош на одредени протоколи. **Оперативните системи можат да бидат различни, потребен е само заеднички протокол** т.е заеднички јазик за комуникација. Корисник на еден компјутер може да се пријави на друг, да преземе некои датотеки итн. Корисникот знае дека не е сам во мрежа т.е свесен е за различните компјутери со кои комуницира преку мрежата.

Дистрибуираните оперативни системи се многу посериозна варијанта во мрежното опкружување. Освен споделување на датотеки овозможува и споделување на процеси т.е програми. Корисниците на овај систем го гледаат како еднопроцесорски систем, но всушност се работи за оперативен систем наменет за работа со повеќе процесори кои се флексибилно поврзани преку мрежа. Тоа значи дека постојат **повеќе компјутери поврзани во мрежа, но само еден оперативен систем управува со сите ресурси во мрежата.**Во вистинскиот дистрибуиран систем, корисникот не треба да води сметка каде се сместени неговите датотеки или каде се извршува неговата програма – тоа е работа на дистрибуираниот оперативен систем. **Дистрибуираниот оперативен систем се однесува како една целина.** Корисникот не мора да знае дека е во мрежа со други компјутери – тој целиот систем го гледа како еден компјутер.

2.Основни функции на ОС:

* Распределба на процесорското време
* Распределба на компјутерската меморија
* Организација на пристапот до надворешните мемории
* Доделување на периферни уреди

3.Што претставуваат регистрите и која е нивната функција?

**Регистер** претставува хардверски уред за чување бинарни податоци. Информацијата се запишува во регистри, се чита од регистри или се пренесува од еден во друг регистер. Регистрите се адресираат од контролната единица на различен начин отколку меморијата.  
**Функција** - се користат за чување адреси, контролна информација или податоци за извршување на програмата.

4. Какви видови на регистри постојат?

Во многу компјутерски системи постојат специјализирани регистри:

* за чување информација, за шифтирање
* споредување вредности, броење
* scratchpad регистри за привремено чување информации
* индексни регистри за контрола на програмски јамки
* stack pointer регистри за работа со стекови со информации за процесите
* статус регистри кои чуваат битови за overflow, carry, zero
* регистри за општа намена достапни на програмерот
* AC (Accumulator) - регистер за општа намена за сместување податоци потребни за работа на CPU
* MAR (Memory Address Register) - за сместување на мемориската адреса на податоците до кои се пристапува
* MBR (Memory Buffer Register) - за сместување на податоци штотуку прочитани од меморија или кои треба да се запишат во меморија
* PC (Program Counter) - за сместување на адресата на следната инструкција која треба да се изврши
* IR (Instruction Register) - за сместување на инструкцијата која треба да се изврши
* InREG (Input Register) - за сместување на податоци од влезните уреди
* OutREG (Output Register) - податоци за сместување во излезни уреди

5.Што претставува Контролната Единица - CU и која е нејзината улога?

**Контролната единица го надгледува извршувањето на сите инструкции и преносот на сите информации.**  CU зема инструкции од меморија, ги декодира, обезбедува да има податоци на вистинско место во вистинско време, и кажува на ALU кои регистри да ги користи, ги сервисира интераптите. CU го користи програмскиот бројач за да најде која е следната инструкција и статус регистерот.

6.Што е магистрала/собирница(bus)?

**Множество линии низ кои битовите се движат паралелно** претставува собирница или магистрала преку која CPU комуницира со другите компоненти. Во еден момент само еден уред (регистер, ALU, меморија) може да ја користи собирницата. Master собирницата ги иницира акциите, додека пак Slave собирницата одговара на барањата.

Тип на собирницата:

* **точка-точка** (point to point) - поврзува две специфични компоненти на системот
* **заеднички пат** (common pathway,multipoint bus) - поврзува голем број уреди овозможувајќи да ја делат собирницата

Во овој случај е потребен bus protocol – множество правила за употреба на собирницата.

7. Од што се состои собирницата?

Типичната собирница се состои од: **податочни линии, адресни линии, контролни линии и линии за напојување.**

* Линиите наменети за пренос на податоци се викаат **data bus** или **податочни линии**.
* Контролните линии одредуваат кој уред има дозвола да ја користи собирницата и за која намена. Контролните линии пренесуваат и потврди за барањата за користење на собирницата, интерапти и сигнали за синхронизација од системскиот часовник.
* **Адресните линии** покажуваат локација од која треба да се читаат податоци или да се запишуваат податоци.

8.Типови на собирници:

* **Процесор-меморија** **собирниците** се многу кратки, брзи и прилагодени да го зголемат преносот на податоци од меморија.
* **I/O собирниците** се подолги и дозволуваат приклучување на различни периферни уреди.
* **Backplane собирницата** е вградена во куќиштето и ги поврзува процесорот, I/O уредите и меморијата.

Персоналните компјутери имаат своја терминологија за собирниците:

* **Интерна собирница (system bus)** која ги поврзува CPU, меморијата и другите интерни компоненти.
* **Екстерна собирница (expansion bus**) која ги поврзува екстерните уреди , периферии, експанзиони слотови и I/O портите со остатокот на компјутерот.
* Повеќето персонални компјутери имаат **локални собирници (local bus)** кои ги поврзуваат периферните уреди директно со CPU.
* Собирниците можат да бидат**: синхрони** и **асинхрони**.

- Кај синхроните собирници преносот се случува под влијание на импулсите на системскиот часовник

- Кај асинхроните собирници контролните линии ги контролираат операциите.

**Кои се типични операции со собирница?**

- испраќање адреса за читање или запис

- трансфер на податоци од меморија до регистер и од регистер до меморија

- читање или запишувања од влезните и излезните периферни уреди

9.Како се врши резервирање на магистрали?

За да може уредот да ја користи собирницата мора да ја резервира, бидејќи само еден уред може да ја користи во даден момент. Само master bus уредите може да ја резервираат собирницата. Тие може да иницираат пренос на информации, а slave bus уредите се активираат од мастер уредот и одговараат на барања за читање или запишување податоци.

Кај системи со повеќе од еден мастер уред потребна е **арбитражна шема** за доделување на приоритет на master bus уредите за користење на собирницата.

Арбитражни шеми за користење на собирницата:

* **Daisy chain arbitration** – преку контролна линија се пренесува дозвола за користење на собирницата од направа со највисок приоритет до направа со најнизок.
* **Централизирана паралелна арбитража** – секоја направа испраќа барање преку своја контролна линија за користење на собирницата, а централен арбитер одлучува кој ќе ја користи собирницата.
* **Дистрибуирана арбитража** – слична шема на претходната но уредите сами одлучуваат кој треба да ја добие собирницата.
* **Дистрибуирана арбитража со откривање на судири** – секој уред испраќа барање. Ако собирницата открие повеќе истовремени барања уредите мораат да испратат други барања за користење на собирницата.

10.Како се врши размена на податоци меѓу процесор и В/И?

CPU разменува информации со влезно/излезните уреди преку влезно/излезни регистри. Размената на информации се изведува на два начина:

* **Мемориски мапирани (memory-mapped) В/И** – В/И регистрите се јавуваат во мемориската мапа на компјутерскиот систем и нема разлика меѓу пристапување до меморија и пристапување до В/И уред. Пристапот е побрз, но се троши меморија.
* **Инструкциски базирани (instruction-based) В/И** - CPU користи специјализирани инструкции за изведување на читање или запис од влезни и излезни уреди. Не се троши меморија, но се потребни специјализирани В/И инструкции.

11. Што претставува виртуелна машина?

Оперативниот систем е дефиниран како Виртуелна машина. IBM развил посебна софтверска структура наречена виртуелна машина: на најниско ниво - хардвер; над него - монитор на виртуелни машини. Оперативниот систем обезбедува т.н виртуелна машина односно единствен поглед на компјутерскиот систем од гледна точка на корисникот, независен од конфигурацијата на компјутерот, конкретниот хардвер и самата архитектура. **Задачата на оперативниот систем како виртуелна машина е да ги работи работите наместо нас** и да ни овозможи некои функции од повисоко ниво на апстракција за полесен пристап кон хардверот.

12. Како се елиминира зависноста од В/И операции?

Бидејќи В/И операциите се многу побавни од процесорот, оперативниот систем мора да ги изолира В/И операции од процесорот, што се постигнува со употреба на брз медиум за привремено зачувување на сите В/И податоци. Тој уред во минатото била лента, а денес по правило е диск. Со цел да се постигне поголема ефикасност , В/И операции по можност треба да бидат што повеќе поклопени, или да се комбинираат со други процесорски работи. Во тој контекст реализирани се две хардверски структури: **канал (Channel )** и **технички прекин (Interrupt).**

* **Канал** - уред кој контролира еден или повеќе периферни уреди, а способен е да пренеси податоци помеѓу периферните уреди и меморијата без интервенции на главниот процесор. Каналот по правило за такви преноси користи услуги на квалитетната хардверска компонента наречена DMA (Direct Memory Access) контролер.
* **Техниката на прекини** - универзална метода за известување на оперативниот систем дека некоја активност т.е В/И команда завршила. Техниката на прекин обезбедува високи перформанси и изолираност од В/И уреди: - DMA ги пренесува податоците по каналот,   
  - Со соодветен прекинувачки сигнал системот се известува дека трансферот е завршен,   
  - Со прекинувачки сигнал управувањето со централниот процесор се пренесува на друга локација.

13. Кои се карактеристиките на оперативните системи?

* **Конкурентност**
* **Поделба на ресурсите**
* **Постоење на долготрајна меморија**
* **Недетерминизам**

14.Поделба на видови на оперативни системи

Критериуми за поделба на оперативните системи се :

* Број на корисници и/или процеси;
* Начин на обработка на податоците;
* Дистрибуција на процесорската снага и останати ресурси;
* Намена;
* Функционални карактеристики;

15. Според начинот на обработка на податоците какви оперативни системи постојат?

* Системи со групна обработка – Корисникот ги предава сите работи за извршување на влезната единица, а тие се извршуваат во низа еден по друг, при што корисникот нема можност да комуницира со својата работа
* Интерактивни системи – преку терминал, timesharing;
* Комбинирани системи – Процесите кои бараат време - пакетно, а другите интерактивно (e-mail)….

16.Какви видови концепции за проектирање на ОС постојат?

Постојат три концепти за проектирање на ОС:

* **Монолитна организација**

Збир на процедури кои по потреба може меѓусебно да се повикуваат без никакви ограничувања.

Не постои никакво групирање или хиерархија.

* **Нивоовска организација**

Во оваа организација ОС се дeли на различни слоеви кои се организирани хиерархиски, односно секој слој може да се повикува само од функциите од пониските слоеви.

* **Архитектура на микројадро**

Основна замисла е да се направи минимално и доверливо јадро со високи перформанси, а сите останати функции на јадрото да се потиснат во т.н. кориснички простор.

16.Како се реализираат системските повици?

Системските повици се реализираат со помош на систем на прекини:

* Корисничката програма ги поставува параметрите на системскиот повик на одредени мемориски локации или регистри на процесорот;
* Го иницира прекинот;
* Оперативниот систем ја презема контролата;
* Ги зема параметрите;
* Ги извршува бараните задачи;
* Резултатот го става на одредени мемориски локации или во регистрите
* Ја враќа контролата на програмата

17.Што претставува јадрото?

Јадрото е основен дел на оперативниот систем, кој овозможува програмата на сигурен начин да му пристапи на хардверот. Јадрото на оперативниот систем не е неопходно за стартување и извршување на програмата. Во хиерархискиот модел јадрото е најблиску до хардверот и ја претставува врската меѓу хардверот и останатите делови на ОС;

18.Функции на јадрото!

Основни функции на јадрото:

* Определување на процес т.е создавање на опкружување во кое може да постојат процеси на процесорите и обезбедување на механизми за интерпроцесна комуникација;
* Како на еден процесор во еден момент може да се изврши еден процес (процесорот е невидлив ресурс);
* Јадрото одредува кога и на кое време процесот ќе добие процесор.

Оваа појава е позната како мултиплексирање и претставува основа кон квазипаралелност (неправилна паралелност).

19.Кои предуслови треба да ги исполни хардверот за јадрото да работи?

За јадрото да ја реализира својата функција неопходно е самиот хардвер да ги исполни одредените предуслови, односно да поседува одредени компоненти. Во тие компоненти спаѓаат:

* **Механизам за прекинување – обезбедува извршување на управувачката програма (прекинувачки способности**) т.е прекинување на контролите за извршување на корисникот и на управувачката програма. Најмалку што механизмот треба да направи е:

**\*да ја сочува вредноста на програмскиот бројач на прекинатата корисничка програма**

**\*да ја покрене управувачката програма** со фиксна локација во меморијата

**\***управувачката програма понатаму одредува извор на прекини и реагира на соодветен начин

* **Заштитен механизам за адресирање на меморија – спречува погрешно адресирање**

**\***т.е спречува еден процес да ги запиши следните податоци во делот на меморијата кој е во другиот процес;

**\***овој процес е автоматски и ги чува и интегрира процесот и податоците кои се наоѓаат во работната меморија;

* **Збир на привилегирани инструкции** – сите инструкции кои се достапни на ОС, но не и на корисничката програма;

\*овие инструкции овозможуваат оперативниот систем да ги сокрива прекините;

\*на процесорот му доделува друг процес;

\*пристапува на регистрите во меморијата

\*извршува влезни-излезни операции;

\*за разлика од непривилегираната инструкција која се извршува во корисничкиот режим, при извршување на привилегирана инструкција оперативниот систем се наоѓа во системскиот режим;

\*корисничката програма (КП) не може да извршува привилегирана инструкција директно, туку само со системски повик;

\*КП со системскиот повик бара од ОС да изврши привилегирана инструкција после што ОС преминува во системскиот режим и ја извршува таа инструкција;

* **Часовник за реално време (real-time clock**) – помага сатниот механизам на реалното време да ги контролира и да ги евидентира потрошените ресурси на компјутерот за сите поединечни процеси.  
  – Овај механизам може да се користи и за распоредување и закажување и извршување на различна работа.

20. Како се дели јадрото според теоретскиот модел?

Јадрото според теоретскиот модел на оперативниот систем (paper model) може да се подели на три основни целини:

* **Прво ниво на обработка на прекини (First Level Interrupt Handler) – FLIH**

- Првото ниво го сочинуваат рутини за одредување на изворот на прекин и иницирање на сервиси, односно опслужување на одредени видови прекини;

- FLIH одговара на надворешните прекини и системски повици;

- После извршувањето на кодот, прекинот се смета за опслужен, а диспечерот одлучува на кој процес ќе го даде процесорот за користење.

* **Диспечер на системот (планер за работи од ниско ниво)**

-Диспечерскиот дел од јадрото кој го доделува процесот на процесорите;

-При тоа процесорот секогаш се доделува врз основа на некој алгоритам како што е Shortest Job First (SJF) –прво работи кои бараат најмалку процесирачко време.

* **Рутини за остварување на интерпроцесни комуникации**

- Рутините за остварување на интерпроцесни комуникации се дел од јадрото на оперативниот систем кој обезбедува комуникација помеѓу процесите;

- Постојат повеќе начини на комуникации како што се:

* Испраќање на порака (send message, post message);
* Семафорски техники;
* Користење на именувани цевки (named pipes, карактериснично за UNIХ);
* Користење на деллива меморија;

21.Што претставува процесот?

**Процесот е програма или дел од програма во состојба на извршување заедно со сите ресурси кои се потребни за работа на програмата**.   
- Кога датотеката ќе се прочита во меморија таа станува процес т.е активен објект кој што има свои ресурси, регистри, меморија; ---Тоа значи доколку три корисници извршуваат некоја активност сите три ќе бидат претставени со различни процеси;   
 Компјутерскиот систем што има само еден процесор може да извршува само една инструкција во еден момент. ОС мора на сите корисници на компјутерскиот систем и сите програми да им обезбеди пристап на процесорот и не смее да пропушти никој да оствари монопол над нив. За таа цел оперативниот систем мора да има одредена информација за секој корисник и своја програма во компјутерот во секој момент. Повеќето оперативни системи управуваат со процесите и задачите.

Процесот претставува еден од најважните концепти на ОС.

22.Од што се состои процесот?

Самиот оперативен процес е истовремено составен од низа процеси. Секој процес има 3 фундаментални мемориски делови т.е секции:

* Програмска или текстуална секција – која не се менува и која содржи програмски код;
* Стек секција – содржи привремени податоци (параметри за процедури, повратни адреси,локални променливи);
* Секција на податоци – содржи глобални променливи;

Како што текстуалната секција не се менува, така поголемиот број оперативни системи формираат заеднички текстуален сегмент за сите процеси кои настануваат после стартувањето на една програма. Овај модел е ефикасен и пожелен бидејќи на тој начин се штеди меморија. Секако дека секој од тие процеси има посебна стек секција и секција на податоци. Процесот опфаќа и :

* Вредност на програмскиот бројач;
* Вредност на останатите важни регистри на процесорот;
* Влезно-излезни ресурси кои што ги користи, влезно - излезни уреди како и разни други датотеки.

23.Што претставува контролен блок на процесот?

Дел од процесот се и **податоците кои се неопходни за управување со процесот.** Овие податоци ги генерира и ги користи ОС, односно диспечерот, а наречени се **КОНТРОЛЕН БЛОК НА ПРОЦЕСОТ (process control block – PCB) вектор на состојби или дескриптор на процесот.** За оперативниот систем да знае каде да го продолжи извршувањето, на секој процес се додава пратечка информација т.е единствен контролен блок.

***Контролниот блок е дел од меморијата односно мемориска структура со основни информации за процесот кој што ОС ги користи за управување со тој процес.***

Во информациите од контролниот блок спаѓаат:

* Име или единствен идентификатор на процесот (PID)
* Контекст (окружување) на процес
* Приоритет на процес
* Моментална состојба на процесот
* Информација за меморијата на процесот
* Листа на отворени датотеки
* Статус на зафатени влезно-излезни ресурси

Дел од контролниот блок во кој се чува контекстот уште се вика и хардверски контролен блок на процесот или хардверски дескриптор на процесот. Контролниот блок се прикажува до самиот код т.е дo процесот, но во реалниот ОС никогаш не се наоѓа во исто мемориско подрачје.

24. Да се објасни животниот век на процесот!

Процесот се состои од низа чекори кои што следат еден по друг. Помеѓу два чекори може да биди прекинат, а неговото извршување да продолжи во друг момент на ист или друг процесор. Носењето на процесот од една во друга состојба го врши оперативниот систем. Процесот може да се најде во неколку состојби ( 5 или 7 зависно од конечниот автомат кој што е користен за опишување на состојбата), а следните три се најважни:

* **Состојба на извршување (RUN, RUNNING)** процесорот извршува инструкции на овој процес;
* **Состојба на чекање на процесор (READY,RUNNABLE)** процесот ги добил сите потребни ресурси освен процесорот и спремен е за работа и да му се додели процесор;
* **Состојба на чекање на ресурс (WAIT,UNRUNNABLE)** процесот чека на некое случување (на пример да се изврши печатење) бидејќи за извршување на процесот е потребен ресурс ( во состојба WAIT чека извршување на влезно-излезните операции или резултат на некои други процеси).

Поради едноставно презентирање се користи конечен автомат во 5 состојби.

* Двете помошни состојби поврзани се со почетокот на процесот, односно моментот на формирање на процесот - **START**.
* И крај на извршувањето на процесот - **STOP**. Процесот во состојба STOP ги ослободува сите ресурси кои ги зел.

25.Во кој случај се јавуваат новите состојби - suspended ready & suspended wait?

Во некои оперативни системи постои можност за привремено прекинување на извршувањето на процесот. Процесот чие извршување е привремено прекинато е суспендиран и престанува да се натпреварува за ресурс. Овај модел на автомати е карактеристичен за оперативниот систем UNIX **. Основна претпоставка е што корисникот кој го иницира овој процес има право во одреден момент да го стопира неговото извршување (suspend)** и исто така тој може да го продолжи извршувањето на суспендираниот процес (revive). **Суспендирањето на процесот може да се јави и како последица на swap просторот**.  
 Извршувањето на процесот може да се прекине само во состојбите WAIT и READY. На тој начин се формираат две додатни состојби:

* **SUSPENDED-READY** во оваа состојба процесот доаѓа доколку е суспендиран во состојба на чекање на процесор;
* **SUSPENDED-WAIT** во оваа состојба процесот доаѓа доколку е суспендиран во состојба на чекање на ресурс;

26.Што претставува ОС што работи со и без претпразнење?

1. Доколку ОС овозможува претпразнење, процесорот може да се одземе до колку наиде процес со повисок приоритет 🡪 со тоа се обезбедува навремено извршување на критични процеси.
2. **Што значи распоредување без претпразнење?**

Процесорот може да се одземе само од процес кој ги завршил своите активности или чека на некој ресурс.

27. Што претставуваат редови на чекање на процесорот и како функционираат?

Процесот во својот живот поминува низ неколку редови на чекање.  
После настанувањето процесот се префрла во **ред на чекање за работа** кој што ги опфаќа сите постоечки процеси на системот. Процесот може да се најде во различни состојби и локации. Сите процеси кои се спремни за работа се наоѓаат во работната меморија и се чуваат за **редот на чекање на процесор** ( во редот на подготвени процеси). Постојат два вида на редови на чекање:

* **Ред на чекање на процесор;**
* **Ред на чекање на влезно-излезен уред;**

Редовите на чекање на процесор по правило се реализираат како поврзани листи на контролни блокови на процеси со дефиниран редослед на извршување на процесорот. Редоследот се задава преку заглавие на листата кое што содржи информација за почетниот и последниот контролен блок во листата и покажувач на следниот контролен блок.   
 Оперативниот систем води посебен ред на чекање за секој влезно-излезен уред. Секој ред на чекање влезно-излезен уред содржи поврзани листи на контролни блокови на процеси кој што тој уред ги бара.

28.Од состојба на извршување во кои други состојби може да се најде процесот?

Процесот кој се наоѓа во состојба на извршување може да:

* Остане без процесор кога ќе истече времето;
* Направи нов процес и да чека во блокирана состојба за да се изврши новиот процес;
* Остане без процес кога ќе настане прекин;
* Постави В/И барање и после тоа да се префрли во ред на чекање за влезно-излезниот уред (да остане блокиран)

Процесот се враќа во ред на чекање за процесор се додека не се заврши, а после тоа ги ослободува сите зафатени ресурси.

29.Што претставува планер на работи?

Програмите за распоредување одлучуваат кога процесот ќе влези во некој ред на чекање или ќе го напушти тој ред. Овие компоненти на ОС се :

* **Планери на работи** ( job scheduler, long-term scheduler, high-level scheduler)
* **Диспечер** (dispatcher, short-term scheduler, low-level scheduler)

Планерот на работите кој во хиерархискиот модел се наоѓа над јадрото ги извршува следните функции:

* *Ги дели работите на процеси;*
* *Врз основа на алгоритми доделува приоритети на процесите;*
* *Ги доведува процесите во ред на чекање за процесор;*

Планерот на работи се повикува само кога ќе се појават нови процеси или кога еден или повеќе процеси завршат. Планерот го регулира степенот на мултипрограмирање.

* Тој прави добра селекција на процеси кои ќе добијат меморија со која системот уште поефикасно би функционирал;
* Првично се појавил кај големите компјутерски системи кои имаат групна обработка на работи;
* Извршувањето на еден процес се состои од:

- еден или повеќе циклуси на извршувања

- еден или повеќе циклуси на чекање на влезно-излезни операции.

* Процесот започнува со циклусот на извршување на процесорот и после тоа може повеќе пати да се менува циклусот или состојбата.

30.Што е диспечер и задача на диспечерот!

**Програмите за распоредување одлучуваат кога процесот ќе влези во некој ред на чекање или ќе го напушти тој ред.**

Задача на диспечерскиот систем е да **доделува процесор на процесите кои што се наоѓаат во процесорскиот ред**. Диспечерот потврдува на кој процес е најповолно да се додели процесорот ( кој што е во состојба READY и има највисок приоритет ). Доколку процесорскиот ред се формира со динамичка листа во која се менуваат приоритетите, најповолен е првиот процес кој што е во редот. Доколку на системот постојат повеќе процесорски редови – по еден ред за секој приоритет - се зема процесот од соодветниот ред. Диспечерот прво одредува дали процесорот треба да му се додели на тековниот процес (тој да продолжи со извршување) или на друг процес.

* Ако продолжува тековниот процес, контролата се враќа на адресата зачувана во прекинувачкиот механизам на јадрото.
* Во случај процесорот да се даде на друг процес во редот, диспечерот најпрво го ажурира контекстот, а потоа врз основа на контекстот на новиот процес поставува окружување во кое тој процес може да се изврши. Потоа контролата на извршување се префрла на она место во новиот процес каде што извршувањето било прекинато.

Планерот и диспечерот може да бидат посебни програмерски модули или еден заеднички.

🡪 Ако се посебни модули - планерот бира процес од редот на чекање и потоа, диспечерот го ажурира контекстот на процесот, поставува окружување и ја предава контролата над процесорот на веќе избраниот процес.

Диспечерот може да се повикува многу често (100мсек). Времето за кое диспечерот застанува процес и му доделува процесор на другиот познато е како каснење на диспечерот.

31.Што претставува замена на контекстот (context switch) ?

– Контекстот го сочинуваат податоци кои се чуваат при одземањето на процесорот и кои овозможуваат продолжување на процесите (регистрите на процесорот, мемориските секции и листата на отворени датотеки).

***Замената на контекстот на процесот претставува преполнување на системот (overhead) но, заради спроведување на мултипрограмирањето мора да се изврши.***

Преполнувањето на системот зависи од:

* Хардверските карактеристики на процесорот
* Бројот на регистри на процесорот кои мора да се чуваат
* Меморијата
* Техниките за управување со меморијата
* Инструкциите кои се задаваат

32.Што е среден распоредувач? Кои се неговите функции?

Употребата на swap техниката во интерактивните системи води до воведување на ниво на распоредување на процесите - **среден распоредувач (intermediate scheduler).**   
Суштината е следна:

* Секој новонастанат процес доаѓа во ред на чекање и после тоа диспечерот оддлучува кому да му го додели процесот;
* Некои процеси можат да бидат суспендирани и запишани на дискот ( во swap просторот) со што се ослободува меморија за другите процеси;
* Кога од дискот ќе се префрлат во меморијата, суспендираните процеси се враќаат во редот на чекање на процесорот;

Средниот распоредувач ги извршува следните функции:

* Функцијата за суспендирање на процесот (swap-out);
* Функцијата на враќање на процесот во состојба подготвеност (swap-in)
* Изборот на процес за двете функции

33.Да се објаснат односите меѓу процесите: процес-дете и процес-родител!

Односите можат да се опишат според делењето ресурси и начинот на извршување:

Според делењето на ресурси, можат да се најдат во следните релации:

* Процесот родител и процесот дете да ги делат ресурсите;
* Процесот родител и процесот дете ги делат подмножеството ресурси на родителскиот процес;
* Процесот родител и дете не ги делат ресурсите;

Особено чуствителен е и меморискиот адресен простор каде се применуваат следните техники:

* Процесот дете го дуплира адресниот простор на родителите;
* Адресниот простор на детето се генерира според програма со која тој адресен простор се полни;

Според начинот на извршување процесот дете со своите родители може да се најде во следните релации:

* Процесот родител продолжува да се извршува независно и конкуретно со детето;
* Процесот родител се блокира и чека се додека процесот дете не заврши со своите активности;

34.Како може да заврши процесот?

* **Нормален крај –** процесот ги завршува своите активности тогаш кога последната наредба на програмата која што се наоѓа во адресниот простор на тој процес ќе заврши; После тоа користејќи го системскиот повик EXIT процесот ќе бара од оперативниот систем да го уништи односно да го избриши. При тоа процесот дете може да ги врати излезните податоци на родителскиот процес со помош на системскиот повик WAIT. Потоа се ослободуваат сите ресурси кои припаѓаат на процесот како што се физичка и виртуелна меморија, датотеки и влезно-излезни уреди;
* **Насилен крај –** Процесот може да предизвика и насилен завршеток на работата на процесот со помош на специјален повик ABORT. По правило процесот родител може да го прекине извршувањето на процесот што го направил, тоа го прави од следните причини:

**-** процесот дете при користењето на некој ресурс ја надминал квотата;

**-** активноста која процесот дете ја извршува не е веќе потребна;

**-** процесот родител ја завршил активноста пред детето, што се смета за ненормална ситуација која оперативните системи не ја дозволуваат - и во тој случај сите процеси деца мора присилно да се уништат (cascade termination).

35.Да се објаснат рутините за интерпроцесна комуникација!

Рутините за интерпроцесна комуникација мора да се имплементирани во јадрото на оперативниот систем кој обезбедува комуникација помеѓу процесите и да бидат достапни за сите процеси и за да имаат директен пристап до диспечерот. Интерпроцесната комуникација може да се оствари на повеќе начини:

* Семафорски техники
* Пуштање на пораки
* Користење делливи мемории
* Именувани цевки

36.Што претставува socket?

**Socket (приклучок) -** најчесто користен механизам на размена на пораки преку мрежа за процеси кои се извршуваат на различни компјутери;

Парот процеси кој сака да комуницира преку мрежа формира приклучоци на секоја страна на мрежата по еден за секој процес;

Два параметри:

* IP приклучок (IP адреса на компјутерот на кој приклучокот се формира);
* Соодветен број на приклуоци;

Генерално socket-от користи клиент-сервер архитектура затоа што серверот ги ослушнува socket-ите и анализира дали стигнало некое барање од клиентите на тој приклучок. Серверите даваат точно одредени услуги кои ослушнуваат и ги пренесуваат податоците на точно дефинирани приклучоци.

37.Што се нитки?

Лесни и тешки процеси:

* **Лесни нитки** (threads) основни целини за извршување на кодот на современите ОС:  
  - **Програмерска целина која треба да заврши една задача:**  
   - имаат свои уникатни ресурси и заеднички ресурси со останатите нитки на истиот процес;  
   - од уникатните ресурси имаат посебен идентификатор на нитката – thread ID посебна вредност на програмскиот бројач, вредностите на другите регистри на процесорот и посебен стек;
* **Тешки нитки** - повеќе нитки припаѓаат на класичен т.е тежок процес. Тој има свој:

- програмски бројач и други процесорски регистри;  
 - три мемориски секции (код, податоци и стек);  
 - влезни-излезни ресурси како што се датотеки и уреди;

**Предности од употребата на програми со повеќе нитки:**

* + **Намалување на времето на одзив** – повеќенитна техника која им овозможува на интерактивните апликации да продолжат со работа, дури и кога дел од програмата е блокиран или извршува некоја долготрајна апликација;
  + **Поефикасно делење на ресурсите и економичност** – на пример, делењето на сегменти на кодот овозможува голем број на нитки со една апликација да се извршува во ист адресен простор. Исто така, изработката на процеси и префрлување на контекстот се поспори од генерирање на нитки и префрлување на контекстот на нитките;
  + **Подобро искористување на повеќепроцесорската архитектура** – било кои нитки може да се извршуваат истовремено, секоја на различен простор.

38.Разлики меѓу корисничките и јадрените нитки!

* За разлика од корисничките нитки, нитките на јадрото не мора да припаѓаат на процес
* Распоредувањето на процесорот за корисничките нитки го извршува библиотека за работа со кориснички нитки, додека пак распоредувањето на јадрените нитки го извршува процесорот
* Јадрото не е свесно за корисничките нитки - потполна поддршка за работа со кориснички нитки обезбедува посебна библиотека

39.Концепции за пресликување на корисничките нитки во нитки на јадрото!

Постојат три концепции за пресликување на корисничките нитки во нитки на јадрото, односно три повеќенитни модели:

* **Повеќе во една** – повеќе кориснички нитки во една јадрена;

- управувањето се одвива на корисничко ниво;

- може да биде ефикасно, но има недостатоци;  
- ако една нитка изврши блокирачки системски повик, ќе се блокира целиот процес, односно сите негови нитки се блокираат;

- само една нитка може да пристапи кон јадрото во еден момент не може да се користи повеќе – процесорска архитектура;

* **Една во една** – секоја корисничка нитка се пресликува во една јадрена;

- конкурентно извршување на нитките;

- карактеристичен за Windows NT, Windows 2000/XP/2003 OS/1<  
- кога една нитка ќе направи блокирачки системски повик останатите може да продолжат;

- повеќе јадрени нитки може да се извршуваат во повеќе-процесорска архитектура;

- ОС обично го ограничува бројот на јадрени нитки;

* **Повеќе во повеќе** – повеќе кориснички нитки се пресликуваат во помал број или ист број нитки во јадрото во зависност од ОС и од бројот на процесори

- најсложен и најквалитетен модел;

- се прилагодува на апликацијата и процесорската околина;

40.Како се креираат/прекинуваат нитки?

Мора да постои повик за креирање на тежок процес кој ќе ги дуплира сите нитки и нивните адресни простори;

* Сигналот упатен кон повеќенитен процес се проследува кон:

- Само нитката за која е наменет;

- Само нитки на процесот;

- Само некои од нитките на процесот;

- Само кон нитката која треба да ги прима и обработува сите сигнали за процесот;

* Прекин на нитка - активност која го прекинува извршувањето на нитката пред нејзиниот природен крај:

- нитката која ќе биде прекината се нарекува ***целна нитка (target thread)***

- може да се случи по две сценарија:  
 🡪 Асинхрон прекин – се прекинува веднаш без оглед во која состојба се наоѓа;

🡪Одложен прекин – целната нитка периодично проверува дали треба да ја прекине активноста и ако треба, штом заврши со некоја целина (изврши дел од работата до крај) ќе прекине.

Нитките ја делат секцијата за податоци во еден процес;

Податоци специфични за нитката – една нитка заради специфични потреби може да направи приватна копија на делови од секцијата за податоци.

41.Дали може да биде прекинат процес во режим на јадрото?

**Процесот кој со системски повик преминал во режим на јадрото не може да биде прекинат се додека е во тој режим.**

42.Во кој случај се доделува процесор на друг процес кој е распределен со претпразнење?

Процесорот се доделува на друг процес под следните услови:

* Кога процесот премине во состојба на чекање на ресурс ( чека на В/И операција која ја иницирал);
* Кога процесот родител чека процесот дете да заврши со активност;
* При транзиција RUN-STOP (кога тековниот процес ги завршил активностите);
* При транзиција RUN-READY;
* При транзиција WAIT-READY (во случај процесот кој ја напуштил состојбата WAIT и преминал во состојба READY е со повисок приоритет од процесот кој тековно се извршува).

43.Во кој случај настанува застој/блокада?

Во повеќепроцесно опкружување, повеќе процеси може да се натпреваруваат за конечен број ресурси. Кога процесот бара ресурс, а ресурсот не е расположлив, процесот влегува во состојба на чекање на ресурс (WAIT) и се блокира. Блокираниот процес може засекогаш да остане во таа состојба до колку не се ослободи ресурсот. Ова може да се јави кога ресурсот е доделен на друг процес кој исто така влегува во состојба на чекање. Ова е состојба на застој или блокада (deadlock).

44.Ако на системот постои само еден процес:

* Не е можно да се појави застој. Ова е директна последица на условите на задржување на ресурсите и чекање на други ресурси.

45.Како изгледа форматот на инструкција?

Инструкцијата се состои од **код на операција** и **информации за тоа каде да се земат операндите и каде да се смести резултатот.**

\*\*\*Нула- адресен, едно-адресен, дво-адресен и три-адресен инструкциски формат.

* Ако сите инструкции имаат иста должина декодирањето е поедноставно, но постои загуба на простор (сите инструкции треба да се долги колку најголемата).
* Инструкциите може да имаат помала , иста или поголема должина од меморискиот збор.
* Според тоа каде се сместени операндите архитектурите се делат на:
* **Меморија-меморија** – може да имаат 2 или 3 операнди во меморија
* **Регистер-меморија** – најмалку еден операнд е во регистер, а другите во меморија
* **Load-Store** – операндите се сместени во регистри
* **Стек** - ако пристапот до меморија е брз стек-базиран дизајн е добра идеја , но ако меморијата е бавна подобро е да се користат повеќе регистри како со UltraSPARC III.

46.Типови на адресирање:

* Непосредно адресирање
* Регистерско адресирање
* Директно адресирање
* Индиректно адресирање
* Индексно и базно адресирање
* Stack адресирање

47.Што претставува меморијата?

Меморијата е логички структурирана како линеарно поле со локации кои имаат адреси од 0 до максималната големина која процесорот може да ја адресира. За изработка на меморија постојано се воведуваат нови технологии со цел да се следат перформансите на CPU.

Има 2 основни типа на меморија:

🡪RAM (Random Access Memory)

🡪ROM (Read Only Memory)

48.Хиерархија на мемории!

Секој тип на меморија има различна брзина на пристап до податоците и различен капацитет. Хиерархијата на мемории ја сочинуваат регистри, кеш, главна и секундарна меморија. Меморијата која е поблиску до процесорот има помало време на пристап до податоците. Мемориите поблиску до врвот на пирамидата се помали, побрзи и поскапи. Кога е потребен податок процесорот испраќа барање до кеш меморијата ( регистрите се за специјална намена). Ако податокот не се пронајде во кеш меморијата барањето се проследува до мемории од пониско ниво на хиерархијата. Кога пониските нивоа одговараат на барањата на повисоките тие испраќаат блок податоци соседни на бараниот податок.

Мемориската хиерархија функционира заради особината локалност:

* **Временска локалност** – податоците кои се користени може повторно да станат потребни во блиска иднина
* **Просторна локалност**  – податоците при работа со полиња или јамки формираат кластери во адресниот простор
* **Секвенцијална локалност –** постои тенденција на инструкциите да им се пристапува секвенцијално

49.Термини од хиерархијата на мемории:

* Погодок (hit)
* Промашување (miss)
* Процент на погодувања (hit rate)
* Процент на промашувања (miss rate)
* Време на обработка на погодок (hit time)
* Време на обработка на промашување (miss penalty)

50.Што е мемориска единица?

**Мемориска единица** *е колекција на ќелии (локации*) кои имаат можност за памтење, заедно со колата кои можат да ја пренесат потребната информација од и кон мемориската единица. Мемориската единицата се специфицира според:

* Бројот на зборови што може да ги содржи
* Бројот на битови што го формираат секој збор

***Капацитет на меморијата*** - вкупниот број на мемориски локации во мемориската единица.

51. Што е управување со меморијата?

Цели:

* **Алокација** на меморијата т.е доделување на меморијата на процесите
* **Разделување на физичкиот и логичкиот простор** на програмата и врзување на адресата
* **Логичка организација** на меморијата, што значи разделување на неизменливите сегменти од податоците

**Релокација** која опфаќа собирање односно дефрагментирање на работната меморија и swap суспендирање на процесот со негово сместување на дискот

* **Подршка за динамичко полнење** на меморија со програмата и динамичко поврзување

52. Доделување на меморија!

Целта е што поефикасно доделување на меморијата. Меморијата се дели на најмалку два дела (две партиции) од кои :

* Едниот (најчесто нискиот дел) е наменет за резидентниот дел на оперативниот систем **(kernel space) -** најчесто се наоѓаат табели за прекинувачки рутини
* Другиот (повисоките делови) е наменет за корисничките процеси **(user space)**

53. Како се формира виртуелен адресен простор?

Збирот на сите логички адреси кои ги генерира програмата се нарекува логички или **виртуелен адресен простор**, а збирот на сите физички адреси кои што нив им одговараат се нарекува физички адресен простор.

Мапирањето (пресликувањето) на виртуелниот адресен простор во физички го прави хардверскиот уред кој се вика MMU (Memory-Menagement Unit) – единица за управување со меморија.

54.Заштита на меморијата!

Заштита на оперативниот систем од корисничките процеси и меѓусебната заштита на корисничките процеси во поглед на пристапот кон мемориските секции може да се реализира со помош на два регистри:

🡪 **релокациски регистер**, кој ја содржи најниската адреса на процесот  
 🡪 **регистер на ограничување**, кој содржи најголем опсег на логички адреси на процесот

Релокацискиот и регистерот на ограничување се два регистри на процесорот кои се полнат тогаш кога процесот добива процесор за извршување.

55.Што значи размена на меморијата?

Некогаш процесот привремено може да се префрли од меморија на диск така што би се ослободила меморија. Слободната меморија се полни со друг процес, после одредено време процесот може да се врати од дискот во меморијата и да продолжи со извршување.

Размената се користи и во приоритетни шеми за распоредување на процеси, каде што процесите од висок приоритет се чуваат во меморијата, додека пак сите процеси со низок приоритет се чуваат на диск и чекаат да се ослободи меморијата. Оваа варијанта на разменување се нарекува roll out, roll in.

Процесот кој се разменува мора да биде потполно ослободен од активностите, не смее да работи, ниту да чека крај на некоја влезно-излезна операција.

Техниката на размена наложува постоење на 3 компоненти:

* Простор на дискот (swap space) на кој ќе се сместат ‘заспаните‘ процеси
* Механизам swap-out кој го префрлува процесот од меморијата на дискот
* Механизам swap-in кој го враќа ‘заспаниот‘ процес од дискот на меморија

56.Програмерски техники за управување со меморијата:

* Динамичко вчитување на програмата во меморија
* Динамичко поврзување
* Техника на преклопување

57. Што е динамичко вчитување?

Сите рутини на програмата се чуваат на дискот во релокатибилен формат. Кога рутината се повика од програмата се проверува дали таа е веќе во меморијата. Ако не е , се повикува полначот да ја вчита. **Суштината на динамичкото вчитување се состои од сместување само на потребните делови на програмата во меморијата, при што се вчитуваат само тогаш кога ќе ги повика програмата.**

Предност на динамичкото полнење е:

* Рутините кои моментално не се потребни не зафаќаат место во меморијата
* Не бара специјална подршка од оперативниот систем - програмерот може сам да ги проектира своите програми
* ОС му помага на програмерот така што обезбедува библиотека за динамичко вчитување
* Овозможува лесна размена на системските библиотеки

58.Што е динамичко поврзување?\*

Некои ОС подржуваат само концепт на статичко поврзување.

* При статичкото поврзување системските библиотеки се третираат како и сите други објектни модули кои се комбинираат со корисничката програма во единствена извршна верзија и со неа се полни меморијата.
* При статичното поврзување, секој повик на системска библиотека бара кодот на комплетната рутина библиотеката да ја копира во кодот на програмата и тоа за секој повик поединечно, со што расте програмата и на дискот но и во меморијата кога ќе дојде време за извршување.

Концептот на динамичко поврзување е сличен на концептот на динамичко полнење. Како што меморијата се полни со модули по потреба исто така се постапува и со поврзување во време на извршување - рутината од системските библиотеки се полни по потреба во време на извршување.

При динамичкото поврзување:

🡪**се прави функција за поврзување**, како дел од кодот за секој повик на системската библиотека

🡪функција за поврзување покажува како да се лоцира соодветната системска рутина и како да се наполни, ако веќе не е во меморија

🡪пред функцијата на поврзување, рутината која што еднаш се доведе во меморијата може да се користи повеќе пати. Значи, за секој повик на иста функција во програмата имаме само едно мемориско полнење и зафаќање.

59.Како функционира техниката на преклопување?

За да се овозможи извршување на процесот кој е поголем и од самата физичка меморија се користи техниката на преклопување. Преклопувањето овозможува во меморијата да се чуваат само оние делови на програмата кои се потребни во тој момент. Кога ќе бидат потребни другите делови тие ќе се вчитаат во меморијата на местото на деловите кои повеќе не се потребни.

Дефинираме две компоненти на преклопување:

* Overlay A(преклопување А), кој се состои од заеднички податоци и рутини и код за првиот дел на програмата кој изнесува 130КБ
* Overlay B(преклопување Б), кој се состои од заеднички податоци и рутини и код за вториот дел на програмата кој изнесува 120КБ

На програмата мора да се додаде драјвер кој ја управува техниката на преклопување.

Техниката на преклопување ја забавува работата на самата програма, бидејќи деловите на програмата се вчитуваат во повеќе итерации, но со неа се постигнува нешто што поинаку не може да се постигне - извршување на програма која е поголема од оперативната меморија. Техниката на преклопување може да биде корисна на системите со ограничени ресурси.

60.Мултипрограмирање со партиции со променлива големина

**Алокацијата на меморија е динамичка** – меморијата се состои од процеси и празнини, а оперативниот систем динамички води евиденција за зафатената меморија на еден од следните начини:

* Бит мапи
* Поврзани листи
* Систем на здружени парови - buddy system

61.Што претставуваат поврзани листи?

Поврзаните листи се градат од записи со следните структури:

* Првото поле на записот означува **ТИП НА МЕМОРИЈА** (P значи дека се работи за процес, а Н дека се работи за слободна меморија)
* Второто поле ја содржи **ПОЧЕТНАТА АДРЕСА** на делот на меморијата кој ја опишува дадениот запис.
* Третото поле ја означува **ДОЛЖИНАТА НА ОПИШАНАТА МЕМОРИЈА**
* Четвртото поле содржи **ПОКАЖУВАЧ** кон следниот запис.

Управувањето со меморијата се одвива на следниот начин:

* При земање на меморијата, во поврзаната листа се бара празнина (запис од типот Н) со доволна големина.
* Ако се најде соодветна празнина, наместо Н се пишува Р, а евентуално се пушта и нов јазол од типот Н (доколку процесот ја зазема цела празнина ќе остани помала празнина)
* Кога процесот А ќе заврши со работа, се ослободува зафатената меморија.

62.Како работи системот на здружени парови?

За управувањето со меморијата се користи по една листа за секој блок слободна меморија со големина 2n В. На почетокот целата меморија е празна така да во листата за празнини со големина 1 МВ имаме еден запис, додека останатите листи се празни.

63.**Страничењето** е метода со хардверска подршка на ниво на процесорот која се користи во сите оперативни системи и на сите компјутерски архитектури.

Физичката меморија т.е физичкиот адресен простор се дели на блокови со фиксна големина кои се нарекуваат **физички страници** или **рамки**.   
 Логичкиот адресен простор исто така, се дели на блокови со иста големина кои се нарекуваат **логички страници (pages).**

На секоја логичка страница одговара една физичка а коресподенцијата помеѓу нив се чува во табела на страници. Во страничењето логичката адреса е еднодимензионална.

64**.Сегментација** - е метод за управување со меморијата кој поддржува логички кориснички поглед врз меморијата.

**Логичкиот адресен простор** се состои од колекција на сегменти, а секој сегмент има единствено име и должина. **Логичката адреса** се состои од два дела:

* **Име на сегментот**
* **Поместување внатре во сегментот**

При сегментација се јавува екстерна фрагментација. Проблемот на екстерната фрагментација, може да се намали со збивање на меморијата. Во методот на сегментација корисничките адреси се дводимензионални, но мора да се транслираат во еднодимензионални физички адреси. Ова мапирање се извршува преку табели на сегменти.

65. Сегментирање со страничење.

Кај процесорите кои имаат вградено подршка и за сегментација и за страничење се овозможува примена на комбинирани методи на дисконтинуирана алокација па така процесите може да се делат на физички дисконтинуирани логички целини. При тоа, страничењето ја поништува екстерната фрагментација.

Логичката адреса се состои од:

* Идентификатор на сегментот (selector)
* Поместување во рамката на сегментот (offset)

Страничењето е реализирано во две нивоа:

* Надворешна табела наречена каталог на страници
* Внатрешна табела - табела на страници

Цел - за да се искомбинираат предностите на двата методи некои процесори и ОС ги страничат сегментите.

Секој процес има:  
 - една табела на сегменти   
 - неколку табели на страници: една табела по сегмент.

Виртуелната адреса се состои од:

* **Број на сегментот** – се користи за индексирање на табелата на сегменти чиј влез ја дава почетната адреса на табелата на страници за тој сегмент
* **Број на страницата** – се користи за индексирање на табелата на страници за да се добие соодветен број на рамката
* **Поместување (офсет)** – се користи за да се лоцира зборот внатре во рамката

66.Функции на влезно-излезен потсистем!

Влезно-излезниот потсистем претставува подсистем кој ќе обезбеди комуникација со влезно-излезните уреди. Влезно-излезниот потсистем е дел од оперативниот систем кој ги извршува следните функции:

* Ги управува влезно-излезните уреди и операции кои тие уреди ги извршуваат и ги контролираат;
* Обезбедува што поедноставен интерфејс кон корисникот и остатокот од системот;

Контролата на влезно-излезните уреди е едно од главните интересирања на проектантите на оперативниот систем. Заради големите разлики во самите уреди потребно е системот да се имплементира со различни методи на контрола на уредите а самата комплексност на управување со уредите мора да биди содржана и скриена во јадрото.

67. Како работи DMA?

**DMA(Direct Memory Access**) се користи за да се избегне програмирањето на В/И за трансфер на големи количини на податоци. Техника на програмиран В/И – кога уредот ги подготвил податоците, процесорот треба да го превземи секој бајт од контролерот. При тоа, процесорот секој пат го проверува статусниот бит, што значи се извршуваат два В/И циклуци.

**DMA контролер** – за да може да се направи брз пренос на податоци кај уредите како што е дискот ( кој има огромно количество податоци) , на релација контролер-работна меморија и обратно

* Сигнали за управување со трансферот од контролерот
* Сигнали за управување со мемориските циклуси
* DMA мора да се програмира
* Компјутерот содржи повеќе DMA канали и секој има свој В/И контролер

Со DMA се бајпасира (одбегнува) процесорот при трансферот меѓу В/И уред и меморијата.

68.Што претставува кеш?

**Кеш (cache)**  претставува област на брза системска меморија која ја чува копијата на податоците, најчесто од дискот. Пристапот на податоците во кешот е значително побрз од пристапот кон податоците на влезно-излезните уреди. Кеширање претставува техника на копирање делови од дискот во кеш меморијата, со што значително се подобруваат перформансите на В/И системите од дискот.  
  
**Разликата** меѓу кешот и баферот се состои во тоа што:

* Баферот моментално ги чува актуелните податоци
* Кешот чува било која копија од дискот
* Ист мемориски простор може да се користи и за баферување и за кеширање

69. Што е спулер?

**Спулер е бафер** кој привремено ги чува излезните податоци наменети за некој неделлив уред (пример печатачот). Истовремено спулерот овозможува пристап до неделливите уреди на следниов начин:

* Процесите ги запишуваат податоците наменети за уредот на диск
* Оперативниот систем управува со спулерот на тој начин што ги опслужува барањата едно по едно. На пример:
  + Секој процес кој сака да испрати нешто за печатење остава свое барање во спулерот на дискот
  + Процесот кој работи во позадина управува со спулерот и печати едно по едно барање

Користењето на спулерот си има свои предности:

* Процесот релативно брзо го поставува своето барање во бафер, а после тоа е слободен понатаму да ги извршува своите активности;
* Неделливите уреди се користат како привидно делливи што овозможува на поголем број процеси истовремено да го користат уредот

Кога не би се користел спулер, процесите најверојатно би требало најпрво да чекаат уредот да се ослободи за да можат да го користат, а потоа повторно би се блокирале и би чекале уредот да ја заврши операцијата која тие ја иницирале.

70.Што е бафер?

**Бафер е дел од меморијата** која функционира на принцип производител-потрошувач и служи за чување на привремени податоци при пренос на податоците меѓу два уреди или помеѓу уред и апликација. Цели:

* Израмнување на различните брзини помеѓу потрошувачот и произведувачот
* Прилагодување на различни количини на трансфер на податоци
* Одржување на семантиката на копирање

71***. Deamon процес*** - е оној кој го покренува јадрото и се извршува во позадина. Пример, lpd (line printer daemon) најчесто се покрева при подигнувањето на системот за да прифати задачи за печатење и да управува со нив. До колку нема барање за печатење, lpd се извршува, но не е активен.

72.Што претставува домен на заштита?

Проблемот на заштита се сведува на контрола на пристапот кон објектите на оперативниот систем: на објектите можат да им пристапат само оние корисници кои се овластени и над објектот можат да се извршат само дозволени операции. Секој домен дефинира множество на објекти и сите операции кои можат да се извршат на тоа множество. Можноста да се изврши операција над објектот се нарекува **право на пристап (access right**). **Доменот претставува колекција на права на пристап на дефинирани парови** (име на објектот, множество на права).

73.Што е матрица на право на пристап?

Заштитата може да се прикаже како **матрица на пристап** (access matrix) во која редовите претставуваат домени, а колоните претставуваат објекти. Елементот на матрицата (i,j) претставува збир на операции кои процесот од доменот Di може да ги изврши над објектот Oj.

Матрицата регулира како процесите кои припаѓаат на различни домени пристапуваат кон објектите во системот. Меѓутоа во вака дефинирана матрица процесите во одредени ситуации можат да преминат од еден домен во друг домен на заштита и со тоа да остварат поголемо право над објектот. Заради тоа се воведува одредена контрола за преминување на процесите од еден во друг домен на заштита. Префрлувањето на процеси од еден домен во друг претставено е со операцијата Switch.

74.Како се имплементира матрицата?

Матрицата на пристап може да се имплементира на 4 начини, во зависност од збирот на домени/објекти кои се објаснети со конкретната матрица:

* **Глобална табела**
* **Листа за контрола на пристапот на објектите**
* **Листа на можности на домените**
* **Механизам на клучеви**

75.Кои се аспектите на безбедноста и како се реализираат?

Апсолутна безбедност не постои, но безбедносните мерки се применуваат на следниве 4 нивоа:

* **Физичко ниво , Мрежно ниво, Ниво на оперативен систем, Човечки фактор;**

76. Кои се најдобрите начини за идентификација?

Еден од главните проблеми со безбедноста на оперативните системи е афтентичноста на корисниците – оперативниот систем на некој начин мора да знае дали се работи за овластениот корисник. Поради тоа секој корисник мора да се идентификува. Идентификацијата се остварува на 3 начини:

* Специјален хардвер како што е клуч или ID картичка
* Внесување на доверливи информации како што е лозинката
* Биолошките атрибути на корисникот ( отпечаток од прстот, потпис,снимање на очната рожница)

77.Кои се методите на напад и да се објаснат!

* **Одбивање на услуга**

**-** DoS предизвикува стопирање на работата на сервисот или програмата така што на другите корисници им се оневозможува да работат со нив. Овај напад може да се изврши на мрежниот слој преку праќање на голем број на SYN пакети (TCP connection request) . Заштитата може да се воспостави преку контролирање на бројот на SYN пакети во единица време. Овие напади можат да се извршат и на апликациски слој преку праќање на специјални командни програми кои ја стопираат работата на програмата.

* **Лажирање (spoofing) na DNS**

**-** При овај начин на напад, напаѓачот ги прати IP адресите во IP пакетите и се претставува како друг компјутер. Бидејќи DNS не го проверува изворот на информацијата, односно од каде таа доаѓа, напаѓачот може да изврши напад на DNS сервисот давајќи му погрешни информации. Најдобра заштита од овај вид на напад е спречување на рутирање на датаграмите со неисправни изворни адреси.

* **Smurf**

**-** Напаѓачот преку интернет им праќа на сите членови на мрежата ICMP ECHO-REQUEST пакети и како одредишна IP адреса ја става адресата на жртвата. Жртвата прима голем број на ECHO-REPLY пакети и најдобар начин на заштита е да се отфрлат ping пакетите кои се испратени до сите.

* **Нушкање (sniffing)**

**-** Метод во кој со специјална програма наречена нушкач (sniffer) се пресретнуваат TCP/IP пакети кои поминуваат низ одреден компјутер и по потреба се прегледува нивната содржина. Доколку во мрежата циркулираат нешифрирани податоци, нушкачот лесно може да дојде до доверливите податоци.

* **Смртоносен ping (Ping-of-Death)**

**-** Праќање на голем број на ICMP REQUEST пакети може да биде фатално за некои оперативни системи. Заштитата се сведува на прифаќање на ограничен број на ICMP REQUEST пакети во единица време или во краен случај забрана за прием на ICMP REQUEST пакети.

* **Скенирање на приклучоци**

**-** Служи за идентификација на отворените приклучоци на жртвата - напаѓачот испраќа SYN или FIN пакети кон опсегот на приклучоци и очекува RST пакети за неактивните приклучоци. Заштитата се постигнува со прифаќање на ограничениот број на SYN пакети во единица време.

78.Што претставуваат програмските закани?\*

Програмата која ја напишал еден корисник (програмер), а со која се служат други корисници претставува потенцијална закана.

* **Тројански коњ**

**-** Тројанските коњи се нелегален сегмент на код подметнат во кодот на програмата со цел да се промени функцијата или однесувањето на оргиналната програма. Тоа се програми кои се преправаат како да се корисни софтверски алатки, но во позадина се инсталира злонамерен и опасен софтвер.Тројанците креираат ‘BackDoor^ (мала, задна врата) и на секого му дозволуваат да ,,помине,, низ неа. Вообичаен тројански коњ се состои од две програми, Client и Server.

* **Замка**

**-** Авторот на програмата може сакајќи или намерно да остави празно место во својот код. Оној кој ги знае тие места може да подметне свој код со тоа да оствари добивка.

* **Преполнување на стекот или баферот**

**-** Најчесто упатуван напад од мрежа при обид за неовластен пристап кон системот.

Овој тип на напад може да го користат и овластените корисници со цел да го измамат системот и да остварат поголеми права. По правило напаѓачот ја користи грешката во програмата, односно недоволното контролирање на раздвојувањето на стекот, податоците и кодот. Тогаш напаѓачот испраќа повеќе влезни податоци отколку што програмата очекува и ја испробува ранливоста на програмата на следниот начин:

* Преполнување на влезното поле, аргументите на командната линија или влезниот бафер се додека не дојде до стекот
* Преку важечка адреса во стекот се запишува адресата од својот код која се сместува во стекот во следниот чекор
* Полнење на дел на стекот со својот код

Во случај на успешен напад, наместо недоволно заштитена програма ќе се изврши илегалниот код вметнат преку преполнување на баферот.

79. Кои се системски закани?  
  
 Многу оперативни системи обезбедуваат механизам за помош каде процесите можат да прават други процеси. Во такви услови можно е злонамерно користење на датотеките и системските ресурси. Два методи со кои тоа може да се постигне се црви и вируси.

80.Кои се техниките за зголемување на сигурноста на системот?

Еден од начините да се зголеми општата безбедност на системите се сведува на периодична проверка на можни безбедносни пропусти во системот. Потребно е да се провери дали постојат:

* Кратки лозинки или лозинки кои лесно се погодуваат
* Опасни програми со доменски бит
* Неавторизирани програми во системскиот директориум
* Неочекувани процеси кои долго време се извршуваат
* Несоодветна заштита за директориумот
* Несоодветна заштита за системскиот директориумот
* Опасни влезови во програмската патека
* Промени во контролните збирови на системските програми

81. Како работат мрежните бариери?

Комуникацијата меѓу два сигурносни домена може да се ограничи и контролира со помош на **мрежни бариери (Firewalls).**

Со помош на мрежните бариери се формираат контролни безбедносни точки на границите на приватните мрежи.

Мрежните бариери на контролните точки ги испитуваат сите пакети кои циркулираат меѓу приватните мрежи и Интернетот. По испитувањето, мрежните бариери ги одобруваат или ги одбиваат обидите за поврзување на приватната и надворешната мрежа. Поедноставно кажано, бариерата го филтрира сообраќајот на релација локална мрежа - интернет.

82.Што значи откривање на напади?

IDS (Intrusion Detection System) цели кон откривање на напади и реакции на обид за напад на системот. Методите за детекција можат да се ккласифицираат во две групи:

* *Детекција која се базира на ознаки* – каде IDS анализира мрежен сообраќај и бира карактеристичен пример кој го открива нападот, како што е типична секвенца на повеќеструко погрешно пријавување.
* *Детекција на аномалии –* каде IDS открива аномалии како што се системски повици со големо количество влезни податоци кои укажуваат на обид за напад преку преполнување на меѓумеморијата-бафер.

83. Како се заштитува системот со криптографија?

Шифрирањето опфаќа математичка постапка за измена на податоците, така што шифрираните податоци односно шифрите можат да бидат прочитани само од страна на корисниците кои рапсолагаат со соодветен клуч. Процесот на шифрирање со математичка функција со одреден клуч го трансформира отворениот текст односно оргиналната порака или датотека, во заштитен односно шифриран текст. Во основа сите криптологаритми подлежат на математички постапки: супституција и пермутација.

84. Од што се состои централниот процесор?

Процесорот се состои од два дела:

**- Податочна патека(datapath):**

1. регистри (единици за сместување на податоци)
2. аритметички и
3. логички единици поврзани со собирници (bus)

**- Контролна единици:** секвенцирање инструкции и контрола на тек на податоци.

1. **Предности од нитки?**

Предноста од употреба на програми со повеќе нитки се:

-намалување на времето на одѕив - повеќенитна техника која им овозможува на интерактивните апликации да продолжат со работа дури и кога дел од програмата е блокиран или извршува некоја долготрајна апликација.

-поефикасно делење на ресурсите и економичност – на пример делење на сегменти на кодот овозможува голем број на нитки од една апликација да се извршува во ист адресен простор. Исто така изработка на процеси и префрлување на контекстот се поспори од генерирање на нитки и префрлување на контекстот на нитки.

-Подобро искористување на повеќепроцесорската архитектура – било кои нитки може да се извршуваат истовремено, секоја на различен процесор.

1. **Кои се основните разлики помеѓу корисничките и јадрените нитки?**

-За разлики од корисничките нитки, нитките на јадрото не мора да се припаѓаат на процес.

-Распоредувањето на процесорот за корисничките нитки го извршува го извршува библиотека за работа со кориснички нитки, додека распоредувањето за нитките на јадрото го извршува процесорот.

-Јадрото не е свесно за корисничките нитки – потполна поддршка за работа со кориснички нитки обезбедува посебна библиотека.

1. **Да се објаснат концептите за пресликување на кориснички во јадрени нитки!**

Постојат три концепти односно три модели:

**-Повеќе во една (повеќе кориснички нитки во една јадрена)**

\*управувањето се одвива на корисничко ниво,може да биде ефикасно но има недостатоци,ако една нитка изврши блокирачки системски повик ќе се блокира целиот процес т.е. сите негови нитки се блокираат,само една нитка може да пристапи кон јадрото во еден момент не може да се користи повеќе-процесорска архитектура,Solaris 2 и библиотеката Green threads

**-Една во една (секоја корисничка нитка се пресликува во една јадрена)**

\*карактеристичен за Windows NT, Windows 2000/XP/2003 OS/2, конкурентно извршување на нитките, кога една нитка направи блокирачки системски повик останатите може да продолжат, повеќе јадрени нитки може да се извршуваат во повеќе-процесорската архитектура, ОС обично го ограничува бројот на јадрени нитки.

**-Повеќе во повеќе (повеќе кориснички нитки се пресликуваат во помал или ист број на нитки во јадрото во зависност од ОС и од бројот на процесори)**

\*најсложен и најквалитетен модел, се прилагодува на апликацијата и процесорската околина, Solaris-2, IRIX, HP-UX, True64 UNIX.

1. **Што значи распоредување со претпразнење ?**

Доколку ОС овозможува претпразнење, процесорот може да се одземе доколку наиде процес со повисок приоритет и со тоа се обезбедува навремено извршување на критични процеси.

1. **Што значи распоредување без претпразнење?**

Процесот може да се одземе само од процес кој ги завршил своите активности или чека на некој ресурс.

1. **Кои се критериуми за доделување процесори?**  
   **\*искористување на процесорот**-процесорот треба да биде постојано зафатен  
   **\*Пропусна моќ на системот**  
   -Број на процеси извршени во единица време  
   -Зависи од хардверските и системските карактеристики  
   **\*Време потребно за комплетирање на процесот**  
   -Вкупната количина на време за кое се извршува поединечен процес  
   **\*Време на чекање**  
   -Вкупното време на чекање во редот на чекање за процесор  
   -Алгоритмите за распоредување влијаат врз времето на чекање.  
   **\*Време на одѕив**-Време за кое после испраќањето на барање се појавуваат првите резултати од извршувањето на процесот.
2. **Што претставува застој и дали можи да се појави ако во системот постои еден процес?**

Во повеќекратното опкружување, повеќе процеси може да се натпреваруваат за конечен број ресурси. Кога процесот бара ресурс, а ресурсот не е расположлив,процесот влегува во состојба на чекање на ресурс(WAIT) и се блокира. Блокираниот процес може засекогаш да остане во таа состојба до колку не се ослободи ресурс. Ова може да се јави кога ресурсот е доделен на друг процес кој исто така излегува во состојба на чекање.  
Ова е состојба на застој или блокада(deadlock). А ако на системот постои еден процес не е можно да се појави застој. Ова е директна последица на условите на задржување на ресурсите и чекање на други ресурси.

1. **Што претставуваа тешки и лесни процеси?**

**-Тешки –**повеќе нитки припаѓаат на еден класичен т.е. тежок процес. Тој има свој:

**-**програмски бројач и други процесорски регистри,

**-**три мемориски секции(код , податоци и стек)

**-**Влезни – излезни ресурси како што се датотеки и уреди.

**Лесни-** основна целина за извршување на кодот кај современите ОС

**-**Програмска целина која треба да заврши една задача

**-**Имаат свои уникатни ресурси и заеднички ресурси со останатите нишки на ист процес.

**-**Од уникатните ресурси имаат посебен идентификатор на нитката (thread ID посебна вредност на програмскиот бројач, вредностите на другите регистри на процесорот и посебен стек.

1. **Да се објасни начинот на работа на процесот родител-дете!**

Процесот во текот на своето извршување може да одговара на системскиот повик и да прави нови процеси. Процесот кој прави нови процеси е родителски процес додека пак направениот процес се вика дете процес. Секој процес дете може и понатаму да направи нови процеси и на тој начин се формира хиерархиска структура на процеси. Односот на процес дете - процес родител може да се опиши според делењето на ресурси и начинот на извршување.

Според делењето на ресурси процесот родител и дете можат да се најдат во следните релации:

-процесот родител и дете ги делат сите ресурси

-процесот родител и дете ги делат подмножеството ресурси на родителскиот процес

-процесот родител и дете не ги делат ресурсите

Според начинот на извршување процесот дете со своите родители може да се најде во следните релации:

-Процесот родител продолжува да се извршува независно и конкуретно со детето;

-Процесот родител се блокира и чека се додека процесот дете не заврши со своите активности;

1. **Во кои случаи процесот родител може да побара прекин на процесот дете?**

**Нормален крај**-Процесот ги завршува своите активности тогаш кога последната наредба на програмата која што се наоѓа во адресниот простор на тој процес ќе заврши.  
-После тоа користејќи го системскиот повик EXIT процесот бара од оперативниот систем да го уништи односно да го избриши. При тоа процесот дете може да ги врати излезните податоци на родителскиот процес со помош на системски повик WAIT. Потоа се ослободуваат сите ресурси кои припаѓаат на процесот како што се физичка и виртуелна меморија,датотеки и влезно-излезни уреди.  
-Ова се нарекува нормално завршување на работата по завршувањето на сите инструкции

**Насилен крај**  
-Процесот може да предизвика и насилен завршеток на работата на процесот со помош на специјален системски повик ABORT.

По правило процесот родител може да го прекине извршувањето на процесот кој што го направил а тоа го прави од следните причини:

-процесот дете при користење на некој ресурс(меморија, системско време) ја надминал квотата

-активноста која процесот дете ја извршува не е веќе потребна

-процесот родител ја заврши активноста пред детето што се смета за ненормална ситуација која оперативните системи не ја дозволуваат – и во тој случај сите процеси деца мора присилно да се уништат.

1. **Рутини за меѓупроцесна комуникација!**

Рутините за интерпроцесните комуникации мора да се имплементирани во јадрото за да бидат достапни за сите процеси и за да имаат директен пристап до диспечерот. Интерпроцесната комуникација може да се оствари на повеќе начини:  
-семафорски техники  
-пуштање на пораки  
-користење деливи мемории  
-Именувани цевки.  
**Испраќање пораки**Комуникациската врска меѓу два процеси за да се разменат пораки.  
Врската која може да биде: директна комуникација или индиректна комуникација преку поштенско сандаче. Пораките се чуваат во мемориски бафери со одреден капацитет.  
**Семафори**:  
Семафор претставува целобројна негативна променлива,чија вредност штити некој ресурс и овозможува комуникација меѓу процесите. Вредноста на семафорот одредува дали ресурсот за кој тој семафор одговара е слободен или не.

1. **Како се реализира индиректна комуникација?**

Правила за реализација на индиректна комуникација се:

-Пораките се испраќаат и примаат преку

**\*порт**

**\*поштенско сандаче** -тоа е објект во кој ОС ги остава пораките од другите процеси, секое сандаче има единствена идентификација.

Врската се воспоставува меѓу парови процеси кои го делат сандачето. Кон врската може да се придружат повеќе од два процеси. Меѓу секој пар процеси може да постојат повеќе различни врски каде секоја одговара на едно сандаче. ОС мора да обезбеди механизми за правење ново сандаче за секој пар процеси кои сакаат да комуницираат.

1. **Што претставуваат мрежни оперативни системи?**

-Мрежно оперативни системи

- Ги карактеризираат компјутерите поврзани во мрежа

-Овие компјутери содржат релативно висок степен на автономија- секој компјутер има свој оперативен систем, и се во можност меѓусебно да разменуваат податоци со помош на одредени протоколи.

-Оперативните системи можат да бидат различни, потребен е само заеднички протокол т.е. заеднички јазик за комуникација.

-Корисник на еден компјутер може да се пријави на друг, да преземе некои датотеки итн.

-Корисникот знае дека не е сам во мрежа т.е. свесен за различни компјутери со кои комуницира преку мрежата.

1. **Дистрибуирани мрежни системи?**

-Дистрибуираните оперативни системи се многу посериозна варијанта во мрежното опкружување

-Освен споделување и миграција на датотеките овозможува и споделување на процеси т.е. програми.

-Корисниците на овој систем го гледаат како еднопроцесорски систем, но всушност се работи за оперативен систем наменет за работа со повеќе процесори кои се флексибилно врзани преку мрежа.

-Тоа значи дека повеќе компјутери поврзани во мрежа, но само еден оперативен систем управува со сите ресурси во мрежата.

-Во вистинскиот дистрибуиран систем, корисникот не треба да води сметка каде се сместени неговите датотеки или каде се извршува неговата програма – тоа е работа на дистрибутивниот оперативен систем.

-Дистрибуираниот оперативен систем се однесува како една целина

-Корисникот не мора да знае дека е во мрежата со други компјутери- тој целиот систем го гледа како еден компјутер.

1. **Од што се состои централниот процесор?**

**-**Процесорот се состои од два дела:

**-**податочна патека(datapath):

1. регистри (единици за сместување на податоци)
2. аритметички и
3. логички единици поврзани со собирници(bus)

-Контролна единици: секвенцирање инструкции и контрола на тек на податоци.

1. **Што претставуваат регистри која е нивната функција и кои видови постојат?**

**-**Регистер претставува хардверски уред за чување на бинарни податоци.

**-**Информацијата се запишува во регистри, се чита од регистри или се пренесува од еден во друг регистар.

**-**Регистрите се адресираат од контролната единица на различен начин отколку меморијата.

**-**функција- се користи за чување на адреси, контролна информација или податоци за извршување на програмата.

-Во многу компјутерски системи постојат специјализирани регистри:

-За чување информација, за шифтирање

-споредување вредности, броење

-scratchpad регистри за повремено чување информации.

-индексни регистри за контрола на програмски јамки

-stack pointer регистри за работа со stack-ови со информации за процесите

-Статус регистри кои чуваат битови за overflow, carry, zero.

-Регистри за општа намена достапни на програмерот.

1. **Кои се функции на контролната единица?**

-Го надгледува извршувањето на сите инструкции и преносот на сите информации

-контролната единица зема инструкции од меморијата ги декодира обезбедува да има податоци на вистинското место во вистинско време и кажува на ALU кои регистри да ги користи ги сервисира интераптите

-контролната единица го користи програмскиот бројач за да најде која е следна инструкција и статусот на регистерот.

1. **Што претставува собирница и од што се состои?**

**-Собирница (Магистрала,BUS)**

**-**CPU комуницира со другите компоненти преку собирници, множество линии низ кои битовите се движат паралелно.

**-**Во еден момент само еден уред ( регистар,ALU, меморија) може да ја користи собирницата.

**Master** собирницата ги иницира акциите, додека slave одговара на барањата.

**-**Тип на собирници:

**-точка-точка(point to point) –**поврзува две специфични компоненти на системот;

**-заеднички пат( common pathway, multipoint bus)-** поврзува голем број уреди овозможувајќи да ја делат собирницата.

-Типична собирница се состои од податочни линии, адресни линии, контролни линии и линии за напојување.

1. **Кои се типични операции со собирница?**

-испраќање адреса за читање или записи

-трансфер на податоци од меморија до регистер и од регистер до меморија

-читање или запишувања од влезните и излезните периферни уреди

Секој трансфер се јавува за време на циклус на собирницата. Тоа е време меѓу два импулси на часовникот на собирницата.

1. **Резервирање на собирницата!**

За да може уредот да ја користи собирницата мора да ја резервира бидејќи само еден уред може да ја користи во даден момент .

Само master bus уредите можат да ја резервираат собирницата. Тие може да иницираат пренос на информации, а slave bus уредите се активираат од мастер уредот и одговараат на барање за читање или запишување податоци Кај системи со повеќе од еден master bus уред потребна е арбитражна шема за доделување на приоритет на master bus уредите за користење на собирницата.

1. **Да се објаснат арбитражни шеми за доделување приоритет на мастер уреди на собирница!**

-**Daisy chain arbitration-** преку контролна линија се пренесува дозвола за користење на собирницата од направа со највисок приоритет до направа со најнизок.

-**Централизирана паралелна арбитража-**  секоја направа испраќа барање преку својата контролна линија за користење на собирницата, а централен арбитер одлучува кој ќе ја користи собирницата.

-**Дистрибуирана арбитража-** слична шема на претходната но уредите сами одлучуваат кој треба да ја добие собирницата.

**-Дистрибуирана арбитража со откривање на судири –** секој уред испраќа барање. Ако собирницата открие повеќе истовремени барања уредите мора да испратат други барања за користење на собирницата.

1. **Функции на оперативен систем!**

Основни функции на ОС:

Распределба на процесорското време

Распределба на компјутерската меморија

Организација на пристапот до надворешните мемории

Доделување на периферни уреди

-Ракување со влезно-излезни операции

-Ракување со грешки и прекини

-Управување со ресурси

-Заштита на ресурсите од злонамерни напади и случајни грешки на корисникот и грешка во корисничките програми и самиот ОС

-Обезбедување на добар интерфејс за операторот и корисникот

1. **Како се врши елиминирање на зависноста на влезно-излезните операции?**

Бидејќи влезно-излезните операции се многу побавни од процесорот, ОС мора да ги изолира влезно-излезните операции од процесорот што се постигнува со употреба на брз медиум за привремено зачувување на сите влезно-излезни податоци. Тој уред во минатото бил лента а денес тоа е по правило диск. Со цел да се постигне поголема ефикасност влезно-излезните операции по можност треба да бидат што повеќе поклопени или да се комбинираат со други процесорски работи. Во тој контекст реализирани се две хардверски структури: канал и технички прекин.

**Канал –** уред кој контролира еден или повеќе периферни уреди а способен е да пренеси податоци помеѓу периферните уреди и меморијата без интервенции на главниот процесор. Каналот по правило за такви преноси користи услуги на квалитетната хардверска компонента наречена DMA (Direct Memory Access) контролер.

**Техниката на прекин** - универзална метода за известување на оперативниот систем дека некоја активност т.е. влезно-излезната команда завршила.

1. Да се објаснат оперативните системи за управување во реално време!
2. **Концепти за проектирање на ОС!**

Постојаттри концепти за проектирање на ОС:  
**\*Монолитна организација**  
-**Збир на процедури кои по потреба може меѓусебно да се повикуваат без никакви ограничувања**.  
-Корисничките програми ги користат сервисите на ОС на следниот начин: параметрите на системскиот повик се сместуваат на одредено место како што се регистрите на процесорот или стек, по што следува повик на јадрото на оперативниот систем(kernel call)  
-Оваа операција го префрлува процесорот од кориснички режим на работа во системски режим и контролата ја предава на ОС.  
-Во системскиот режим на работа се достапни некои команди на процесорот до кои не може да се пристапи од кориснички режим.  
-После повикувањето на јадрото,ОС ја презема контролата и без основа на параметрите на повикот одредува која системска процедура треба да се повика.  
-После извршувањето на процедурата,контролата се враќа кон корисничката програма  
-ОС ја има следната структура  
-Главна програма која ги обезбедува системските повици.  
-Збир на системски процедури кои се повикуваат со системски повици.  
-Збир на помошни процедури кои ги користат системските процедури.  
**Не постои никакво групирање или хиерархија**  
**\*Нивоовска огранизација  
-Во оваа организација(layered system),ОС се дали на различни слоеви кои се организирани хиерархиски,односно секој слој може да се повикува само функциите од пониските слоеви.**-Првиот нивоовски (слоевит) оперативен систем THE (Technische Hogeschool Eindhoven) се состоел од 6 слоеви. Од 3 до 3 го претставуваат јадрото на ОС и работат во системски режим  
0-управува со процесорот  
1-управува со меморијата  
2-меѓупроцесна комуникација и команден интерпретер  
3-ВИ операции  
4-кориснички програми(не се грижат за сите претходни активности)  
Кај нивоовската организација -секој слој има точно одредена функција и се потпира врз пониските слоеви до кои пристапува со помош на повици слични на системските повици  
-Најнискиот слој е хардверот(ниво 0)  
-Највисокиот – корисничкиот интерфејс(ниво N)  
-Нивоовскиот ОС дозволува повикување(користење) на операциите за создавање и уништување на процеси само од слојот кој во хиерархијата се наоѓа над сите своеви на оперативниот систем. **\*Архитектура на микро јадро  
-Основна замисла е да се направи минимално и доверливо јадро со високи перформанси,а сите останати функции на јадрото да се потиснат во т.н. кориснички простор(user space).**-Корисничките модули комуницираат меѓусебно со испраќање на пораки  
-Карактеристики на микројадрото  
\*Едноставно проширување и оптимизација на јадрото  
\*Едноставно додавање на нови модули без влијание на основното јадро  
\*Едноставна преносливост на друга компјутерска архитектура  
\*Поголема доверливост  
\*Поголема сигурност

1. **Како се реализираат системски повици?**

**Системски повици –** апликативните програми комуницираат со ОС со помош

на системски повици, т.е. преку операции кои ги дефинира ОС .

**Системските повици се реализираат со помош на систем на прекини:**

-корисничката програма ги поставува параметрите на системскиот повик на одредени мемориски локации или регистри на процесорот

-го иницира прекинот

-оперативниот систем ја презема контролата

-ги зема параметрите

-ги извршува бараните задачи

- резултатот го става на одредени мемориски локации или во регистрите и

-ја враќа контролата на програмата.

1. **Што е јадро?**

Јадрото е основен дел од оперативниот систем кој што овозможува програмата на сигурен начин да му пристапи на хардверот. Јадрото на оперативниот систем не е неопходно за стартување и извршување на програмата. Во хардверскиот модел јадрото е најблиску до хардверот и ја претставува врската меѓу хардверот и останатите делови на ОС.

1. **Која е основна функција на јадрото?**

Основната функција на јадрото е:

-определување на процес т.е. создавање на опкружување во кое може да постојат процеси на процесорите и обезбедување на механизми за интерпроцесна комуникација.  
-Како на еден процесор во еден момент може да се изврши еден процес (процесот е невидлив ресурс)-Јадрото одредува кога и на кое време процесот ќе добие процесор.  
Оваа појава е позната како мултиплексирање и претставува основа кон квазипаралелност(неправилна паралелност).

1. **Кои предуслови треба да ги исполнува хардверот за да можи јадрото да функционира?**

За да јадрото ја реализира својата функција неопходно е самиот хардвер да ги исполни одредените предуслови,односно да поседува одредени компоненти. Во тие компоненти спаѓаат:

**-механизам за прекинување –** обезбедува извршување на управувачката програма (прекинувачики способности)

Најмалку што механизмот треба да направи е:

**\*да ја сочува вредноста на програмскиот бројач на прекинатата корисничка програма**

**\*да ја покрене управувачката програма** со фиксна локација во меморијата

**\***управувачката програма понатаму одредува извор на прекини и реагира на соодветен начин

**-заштитен механизам за адресирање на меморија** – спречува погрешно адресирање т.е. спречува еден процес да ги запиши следните податоци во делот во меморијата кој е во другиот процес. Овој процес и ги чува и ги интегрира процесот и податоците кои се наоѓаат во работната меморија.

**-збир на прилевигирани инструкции**- сите инструкции кои се достапни на ОС но не и на корисничката програма. Овие инструкции овозможуваат ОС да ги сокрива прекините, на процесот му доделуваат друг процес, пристапуваат на регистри во меморијата и извршуваат влезно-излезни операции.

\*за разлика од непривилегираната инструкција која се извршува во корисничкиот режим, при извршување на привилегирана инструкција оперативниот систем се наоѓа во системскиот режим;

\*корисничката програма (КП) не може да извршува привилегирана инструкција директно, туку само со системски повик;

\*КП со системскиот повик бара од ОС да изврши привилегирана инструкција после што ОС преминува во системскиот режим и ја извршува таа инструкција;

-**часовник за реално време**- помага сатниот механизам на реално време да ги контролира и да ги евидентира потрошените ресурси на компјутерот за сите поединечни процеси. Овој механизам може да се користи и за распоредување и закажување и извршување на различна работа.

1. **Кои се делови на јадрото според теоретскиот модел на ОС?**

Јадрото според теоретскиот модел на ОС може да се подели на три основни целини и тоа:

\*прво ниво за обработка на прекини – ова ниво го сочинуваат рутини за одредување на изворот за прекин и иницирање на сервиси односно опслужување на одредени видови прекини.

-FLIH одговара на надворешните прекини и системските повици-после извршувањето на кодот прекинот се смета за опслужен а диспечерот одлучува на кој процес ќе го даде процесорот за користење.

**\*диспечер на системот**

-Диспечерскиот дел од јадрото кој го доделува процесот на процесорите , притоа процесорот секогаш се доделува врз основа на некој алгоритам како што се Shortest Job First ( прво работи кои бараат најмалку процесирачко време)

**\*рутини за остварување на интерпроцесна комуникација –** се дел од јадрото на ОС кој обезбедува комуникација помеѓу процесите. Постојат повеќе начини за комуникација како што се:

**-**испраќање пораки

**-**семафорски техники

**-**користење на именувани цевки

**-**користење на делива меморија

1. **Што е процес?**

-Процесот е програма или дел од програмата во состојба на извршување заедно со сите ресурси кои се потребни за работата на програмата.

-Повеќето оперативни системи управуваат со процесите и задачите.

-Процесот претставува еден од најважните концепти на ОС.

-пасивен објект- односно датотека на диск. Кога датотеката ќе се прочита таа станува процес т.е. активен објект кој што има свои ресурси,регистри и меморија. Тоа значи доколку три различни корисници извршуваат некоја активност сите три ќе бидат претставени со различни процеси.

1. **Што е овозможено со контролниот блок?**

**-За да оперативниот систем знае каде да го продолжи извршувањето, на секој процес се додава пратечка информација т.е. единствен контролен блок.**

**-**Контролниот блок е дел од меморијата односно мемориска структура со основни информации за процесот кој што ОС го користи за управување со тој процес.

**-**Благодарение на контролниот блок извршувањето на програмата може да се прекинува и повторува повеќе пати.

Во информациите од контролниот блок спаѓаат:

-име или единствен идентификатор на процесот.

-Контекст( окружување) на процес

-информација за меморијата на процесот

-листа на отворени датотеки

-статус на зафатени влезно излезни ресурси.

-моментална состојба на процесот.

-**Генерално речено, -**процесот го прават податоци кои се чуваат при одземање на процесорот,а нив ги генерира самиот хардвер: програмски бројач, вредност на регистрите и покажувачите на дел од меморијата која што процесот ја користи.

1. **Животен век на процеси?**

**Од книгата 4 то предавање**

1. **Хардверски контролен блок?**

-Делот на контролниот блок во кој се чува контекстот уште се вика и **хардверски контролен блок на процесот**  или хардверски дескриптор на процесот. Контролниот блок се прикажува придружен до самиот код т.е. до процесот, но во реалниот ОС никогаш не се наоѓа во исто мемориско подрачје.

1. **Да се објасни основниот дијаграм на состојби на процесите!**

Процесот се состои од низа чекори кои што следат еден по друг. Помеѓу два чекори може да биди прекинат, а неговото извршување да продолжи во еден друг момент на ист или друг процесор .Сите процеси кои влегуваат во компјутерскиот систем проаѓаат низ низа состојби во текот на престојот во компјутерон. Едноставно речено состојбата на процесот го опишува точно она што се случува во процесот. Носењето на процесот од една состојба во друга го врши оперативниот систем. Процесот може да се најде во неколку состојби(5 или 7 зависно од конечниот автомат кој што е корисен за опишување на состојбата),а следните три се најважни  
**состојба на извршување(**RUN,RUNNING) процесот извршува инструкции на овој процес.  
-состојба на чекање(READY,RUNNABLE)процесот ги добил сите потребни ресурси освен процесорот спремен е за работа да му се додели процесор.  
-состојба на чекање(WAIT,UNRUNNABLE)процесот чека некое случување(на пример да се изврши печатење)бидејќи за извршувањето на процесот е потребен ресурс(со состојба WAIT чека завршување на влезно-излезните операции или резултат на некои други процеси).

1. **Да се објасни проширениот дијаграм на состојби?**

Бидејќи кај некои ОС можно е да се појави привремено прекинување на извршувањето на процесот се појавуваат нови додатни состојби

Suspended-ready (во оваа состојба процесот доаѓа доколку е суспендиран во состојба на чекање на процесор.)

Suspended-wait (во оваа состојба процесот доаѓа доколку е суспендиран во состојба на чекање на ресурс)

На овој начин се добива нов конечен автомат со седум состојби и три нови транзиции. Новите транзиции се:

-**READY-SUSPENDED READY** доведување на процесот од состојба READY во суспендирана состојба.

-**WAIT-SUSPENDED WAIT** го доведува процесот во суспендирана состојба од состојба Wait.

-**SUSPENDEN WAIT-SUSPENDED RAEDY**  ресурсот неопходен за понатамошно извршување на процесот е слободен но процесот и понатаму е суспендиран

**-SUSPENDED READY-READY** процесот е слободен и се донесува на крајот на процесорскиот ред. Оваа транзиција е можна само на експлицитно барање на корисникот

**-SUSPENDED WAIT-WAIT** – процесот е слободен но ресурсот неопходен за негово извршување не е слободен. Оваа транзиција е можна само на експлицитно барање на корисникот.

1. **Кои се механизми за доделување на процесори на различни процеси**

На еднопроцесниот компјутерски систем во еден момент може да се извршува само еден процес. Но на повеќепрограмскиот систем во еден момент можат да се извршуваат толку процеси колку што во системот има процесори. Доколку на системот се извршува повеќе процесен ОС, ОС треба да обезбеди соодветен механизам за доделување процесори на различни процеси. Делови на тие механизми се:

-редови на чекање на процесор и

-планерите на работа

1. **Какви видови редови на чекање постојат/**

Постојат два вида на редови на чекање:

-ред на чекање на процесор- по правило се реализираат како побрзани листи формирани од контролни блокови на процеси со дефиниран редослед на извршување на процесорот. Редоследот се задава преку заглавие на листата кое што содржи информација за почетниот и последниот контролен блок во листата и покажувач не следниот контролен блок.

-ред на чекање на влезно-излезен уред – ОС води посебен ред на чекање за секој влезно-излезен уред. Секој ред н чекање уред содржи поврзани листи на контролни блокови на процеси кој што тој уред ги бара.

1. **Да се објасни редот на чекање на процесор?**

Редовите на чекање на процесор по правило се реализираат како побрзани листи формирани од контролни блокови на процеси со дефиниран редослед на извршување на процесорот. Редоследот се задава преку заглавие на листата кое што содржи информација за почетниот и последниот контролен блок во листата и покажувач не следниот контролен блок. Процесот кој се наоѓа во состојба на извршување може да:

-остане без процесот кога ќе истече времето

-да направи нов процес и да чека во блокирана состојба за да се изврши новиот процес

-остане без процес кога ќе настане прекин

-постави ВИ барање и после тоа да се префрли во ред на чекање за

влезно-излезниот уред.

Процесот се враќа во ред на чекање за процесор се додека не се заврши а после тоа ги ослободува сите зафатени ресурси.

1. **Што е планер на работа и неговите функции?**

**-**Програмите за распоредување одлучуваат кога процесот ќе влези во некој ред на чекање или ќе го напушти тој ред.

**-**Основни компоненти на ОС се нарекува:

**-Планер на работи**

**-Диспечер**

**Планерот на работите** кој во хиерархиски модел се наоѓа над јадрото ги извршува следните функции:

-Ги дели работите на процеси

-Врз основа на алгоритми доделува приоритети на процесите

-Ги доведува процесите во ред на чекање за процесор

Планерот на работи се повикува **само кога ќе се појават нови процеси или кога еден или повеќе процеси завршат.** Планерот го регулира степенот на мултипрограмирањето т.е. број на симултани процеси во меморијата.

-Тој прави добра селекција на процеси кои ќе добијат меморија со која системот уште поефикасно би функционирал.

-Првично се појавил кај големите компјутерски системи кои имаат групна обработка на работи.

1. **Што е диспечер и коа е неговата функција?**

-Задачата на диспечерскиот систем е да доделува процесор на процесите кои што се наоѓаат во процесорскиот ред.

-Задача на диспечерот е да потврдува на кој процес е најповолно да се додели процесорот ( кој што е во состојба READY и има највисок приоритет).

-Диспечерот прво одредува дали процесорот треба да му се додели на тековниот процес или на некој друг процес:

-Ако продолжува тековниот процес, контролата се враќа на адресата зачувана во прекинувачкиот механизам на јадрото.

-Во случај процесорот да се даде на друг процес во редот, диспечерот најпрво го ажурира контекстот, а потоа врз основа на контекстот на новиот процес поставува окружување во кое тој процес може да се изврши. Потоа контролата на извршување се префрла на она место во новиот процес каде што извршувањето било прекинато.

1. **Што е каснење на диспечерот?**

Диспечерот може да се повикува многу често. Времето за кое диспечерот застанува процес и му доделува процесор на другиот познато е како каснење на диспечерот.

1. **Да се објасни замената на контекстот на процесот или преполнување на системот!**

Замена на контекстот на процесот- Контекстот го сочинуваат податоци кои се чуваат при одземање на процесорот и кои може да овозможат продолжување на извршувањето на процесите (регистрите на процесорот, мемориските секции и листата на отворени датотеки). Замена на контекстот претставува чиста загуба на време т.е. преполнување на системот но заради спроведување на мултипрограмирањето мора да се изврши.

Преполнувањето на системот зависи од:

-хардверските карактеристики на процесорот

-меморијата

-бројот на регистри на процесорот

-инструкции кои се задаваат

-техниките за управување со меморијата

Ова преполнување е многу помало при користење на нитки.

1. **Што е средно ниво на распоредување на процесот?**

**-**Употребата на Swap техниката во интерактивните системи води до воведување на ниво на распоредување на процесите- **среден распоредувач.** Суштината на ова е следна:

**-**Секој ново настанат процес доаѓа во ред на чекање и после тоа диспечерот одлучува кому да го подели процесот.

**-**Некои процеси можат да бидат суспендирани ( привремено прекинати) и запишани на дискот ( во swap просторот) со што се ослободува меморија за другите процеси.

**-**Кога од дискот ќе се префрлат во меморијата, суспендираните процеси се враќаат во редот на чекање за процесот.

**Средниот распоредувач ги извршува следните функции:**

-Функцијата на суспендирање на процесот (swap-out);

-Функцијата на враќање на процесот во состојба подготвеност (swap-in) и

-Изборот на процес за двете функции.

1. Што преставува управување со меморијата и кои се целите?

Цели на управување со меморијата се:

-Алокација на меморијата т.е. доделување на меморијата на процесите

-Разделување на физичкиот и логичкиот простор на програмата и врзување на адресата (превод на релативни релокатиблини адреси во фиксни)

-Логичка организација на меморијата што значи разделување на неизменливите сегменти (модули и процедури) од сегментот со променлива содржина т.е. податоците

-Релокација која опфаќа собирање т.е. дефрагментирање на работната меморијата (врзување на одреден број дисконтинуирани мемориски блокови во еден континуиран нефргментиран простор) и swap суспендирање на процесот со негово сместување на дискот

- Поддршка за динамичко полнење на меморијата со програмата и динамичко поврзување

1. Што претставува мемориска единица и како функционира?

**- Мемориска единица** е колекција на ќелии кои имаат можност за памтење, заедно со колата кои можат да ја пренесат потребната информација од и кон мемориската единица.

Мемориската единица се специфира според:

-бројот на зборовите што може да ги содржи и

-бројот на битови што го формираат секој збор.

**Капацитет на меморијата -** вкуппниот број на мемориски локации во мемориската единица.

Комуникацијата со меморијата се изведува преку влезни и излезни податочни линии,адресни линии и контролни линии. Постојат n влезни и исто толку излезни податочни линии.Преку нив се пренесуваат податоците во и од меморијата.

1. **Како се врши доделување на меморија?**

Техники за доделување на меморијата грубо може да се поделат на два вида:

**\*Континуирана алокација** - Логичкиот и физичкиот адресен простор на процесот се состои од низа на мемориски адреси при што мемориските делови кои се доделуваат на процесите може по големина да бидат еднакви или различни

**\*Дисконтинуирана алокација** - Генерално гледано физичкиот адресен простор на процесот не е реализиран како континуирана низа на мемориски адреси . Дисконтинуираната алокација ги опфаќа методите на страничење, сегментација и страничење со сегментација.

1. **Да се објасни врзување на адресите!**

По правило програмата е наоѓа на дискот како бинарна извршна датотека. Програмата мора од дискот да се вчита во меморијата внатре во адресниот простор во новосоздадениот процес. Во зависност од методата за управување со меморијата која што се користи во текот на извршувањето процесот може повеќе пати да се поместува на релација д иск -меморија. Колекцијата процеси на дискот која што чека враќање во меморијата и продолжување на извршувањето се вика влезен ред. Во системот со овозможено мултипрограмирање голем број на процеси ја делат работата на меморијата на компјутерот. Програмерот не може однапред да одреди фиксни мемориски локации за сместување на програмата и затоа користи релативни или симболички адреси. ОС треба да ја преведе релативната адреса во фиксна при вчитување на програмата во меморијата т.е. да ги поврзе адресите.

1. **Како е адресиран адресниот простор?**

-Збирот на сите логички адреси кои ги генерира програмата се нарекува логички или виртуелен адресен простор

- Збирот на сите физички адреси кои што нив им одговараат се нарекува физички адресен простор.

Мапирањето (пресликувањето) на виртуелниот адресен простор во физички го прави единица за управување со меморија која се вика MMU (Memory - Management Unit).

Шемите за мапирање можат да бидат многу сложени.

1. **Како работи релокациски регистер?**

Релокациски регистер ја дефинира адресата на физичкиот почеток на програмата. Секоја логичка адреса која ја генерира програмата се собира со вредноста на релокацикиот регистер и така се добива физичка адреса. Корисничката програма секогаш почнува од нулта адреса и не треба да води грижа за својот физички простор освен за горната граница на програмата (max). Логичкиот адресен простор кој е наоѓа во опсег [0,max] се мапира во опсег

[R + 0. R+max] каде R е вредноста на релокацискиот регистер т.е. физичката адреса на почетокот на програмата.

1. **Да се објасни заштитата на меморијата о релокациски регистер!**

**-**Заштита на оперативниот систем од корисничките процеси и меѓусебната заштита на корисничките процеси во поглед на пристапот кон мемориските секции може да се реализира со помош на два регистри:

**-релокациски регистер** кој ја содржи најниската адреса на процесот

**-регистер со ограничување** кој содржи најголем опсег на логички адреси на процесорот

Релокацискиот регистер и регистерот со ограничување се два регистри на процесорот кои се полнат тогаш кога процесот добива процесор за извршување.

Од контекстот на процесот диспечерот ги чита вредностите на овие два регистри и ги поставува во случај на активирање на процесот. Со помош на овие два регистри, процесите успешно ги штитат своите мемориски секции бидејќи секоја логичка адреса е релативна во однос на нулата и мора да помине на обработка преку овие два регистри.

-**Транзитен код –** делови на оперативниот систем кои не се потребни и кои може привремено да се исфрлат или да се доведат повторно во меморијата кога ќе бидат потребни.

1. **Кои се програмите за управување со меморијата и да се објасни едно или да се објасни динамичкото вчитување на програмата?**

**-Динамичко вчитување на програмата во меморијата**

**-Динамичко поврзување**

**-Техника на преклопување**

**Динамичко вчитување-** сите рутини на програмата се чуваат на дискот во релокатибилен формат. Кога рутината се повика од програмата се проверува дали таа е веќе во меморијата ако не е се повикува полначот да ја вчита во меморијата. Суштината на динамичкото полнење се состои од сместување само на потребните делови на програмата во мемoријата при што се вчитуваат само тогаш кога ќе ги повика програмата.  
\*Предности на динамичкото полнење се:  
-рутини кои моментално не се потребни не зафаќаат место во меморијата што е добро за големи програми.  
-исто така динамичкото полнење не бара специјална поддршка од оперативниот систем - програмерот мора сам да ги проектира своите програми така што ќе ги користи принципите на динамичко полнење.  
-Оперативниот систем може да му помогни на програмерот така што обезбедува библиотека за динамичко полнење.  
-динамичкото полнење овозможува лесна измена на системските библиотеки, а старите програми не мораат да се преведуваат повторно.

1. **Да се објасни динамичко поврзување!**

Некои оперативни системи поддржуваат само концепт на статичко поврзување.  
-при статичко поврзување секој повик на системска библиотека бара кодот на комплетната рутина библиотеката да ја копира во кодот на програмата и тоа за секој повик поединечно, со што расте програмата на дискот но и во меморијата кога ќе дојде времето за извршување.  
-концептот на динамичко поврзување е сличен на концептот на динамичко полнење. Како што меморијата се полни со модули по потреба исто така се постапува и со поврзување во време на извршување - рутината од системските библиотека се полни по потреба во време на извршување.  
-При динамичко поврзување:  
-се прави функција за поврзување како дел од кодот за секој повик на системската библиотека.  
-функцијата за поврзување покажува како да се лоцира соодветната системска рутина и како да се наполни ако веќе не е во меморијата.  
-преку функцијата за поврзување рутината која што еднаш се доведе во меморијата може да се користи повеќе пати. Значи за секој повик на истата функција во програмата имаме само едно мемориско полнење и зафаќање.  
-за разлика од динамичкото полнење динамичкото поврзување бара поддршка на ниво на оперативниот систем - истата системска рутина која е во меморијата може да ја користат повеќе процеси.

1. **Да се објасни техника на преклопување!**

За да се овозможи извршување на процесот кој е поголем и од самата физичка меморија се користи техниката за преклопување. Преклопувањето овозможува во меморијата да се чуваат само оние делови на програмата кои се потребни во тој момент. Кога ќе бидат потребни другите делови тие ќе се вчитаат во меморија на местото на делови кој повеќе не се потребни.

1. **Да се објасни мултипрограма со фиксни партиции!**

Еден од најстарите и наједноставни методи на доделување на меморијата е делење на целата физичка меморија на повеќе делови со фиксна големина при што во еден дел може да се најде само еден процес. Во ваква организација степенот на мултипрограмирање е еднаков на бројот на мемориски партиции. Цела меморија се дели на повеќе делови а процесот се сместува во оној дел кој е еднаков или поголем од меморијата која ја бара процесот. Сите процеси се ставаат во ред на чекање кој може да е единствен за цел систем или посебен за секоја партиција. Доколку за процесот кој е стигнат на ред нема доволно меморија се зема следниот помал процес од листата. Кога постојат повеќе редови на чекање поголемиот број на мали процеси може да почека во редот на мали партиции, додека големите партиции не се искористени. Во тој случај има доволно меморија но не се користи но од една страна подобар е едноставниот ред на чекање бидејќи ако нема место во меморијата во редот на чекање за партиција која одговара на големината процесот се доделува на поголема партиција. Два процеси никако не смеат да бидат сместени во една партиција. Двата вид на ред за чекање се карактеризираат со интерна фрагментација и деловите на меморијата кои се поголеми од процесот се потполно неискористени.

1. **Да се објасни динамичка алокација на меморијата или што е алокација на меморијата со ВРМ?**

**Алокацијата на меморијата е динамичка-** меморијата се состои од процеси и празнини, а оперативниот систем динамички води евиденција за зафатената меморија на еден од следниве начини:

**\*Бит мапи**- Меморијата се дели на делови со иста големина а во крајни случаи тие се бајти. На секој дел му доделуваме по еден бит на следниот начин: 1 означува дека тој дел е зафатен а 0 дека е слободен или обратно. Така доаѓаме до низа од нули и единици кои ја претставуваат бит мапата на меморијата.

**\*Поврзани листи**- Поврзаните листи се градат од записи со следните структури: Првото поле на записот означува тип на меморијата (Р значи дека работи за процес Н значи дека се работи за слободна меморија) Второто поле ја содржи почетната адреса на делот на меморијата која ја опишува дадениот запис. Третото поле ја означува должината на опишаната меморија. Четвртото поле содржи покажувач кон следниот запис.

1. **Линкирани листи!**
2. **Да се објасни алгоритам за додавање на партиции!**

Алгоритми за додавање на партиции се:

1. **Што претставува интерна фрагментација, а што екстерна и како се решава проблемот со екстерна?**

**Интерна фрагментација –** се јавува доколку на процесот му се додели мемориска партиција поголема од меморијата која ја бара процесот. Останатата меморија е неупотреблива.

**Екстерна фрагментација** - меморискиот простор не е континуиран, така што не може да се задоволи барањата на процесите иако останало доволна меморија.  
Проблемот со екстерна фрагментација се решава со собирање(стеснување)на меморијата за да се добие континуиран слободен простор. Тоа не е секогаш можно, собирањето е можно ако релокацијата на програмата е динамичка и ако се извршува во време на извршувањето. Собирањето ги деградира перформансите на системот бидејќи системот мора да ги прекинува процесите, привремено да ги префрли на диск.  
Најдобро решение - дисконтинуирана алокација на меморија.

1. **Што претставува страничење и како функционира?**

Страничењето е метода со хардверска поддршка на ниво на процесорот која се користи во сите оперативни системи и на сите компјутерски архитектури. Физичката меморија т.е. физичкиот адресен простор, се дели на блокови со фиксна големина, ко се нарекуваат физички страници или рамки (page frames).Локалниот адресен простор исто така,се дели на блокови со иста големина кои се нарекуваат логички страници (pages). Под терминот страница се подразбира логичка страница ,а под рамка физичка. Методата на страничење функционира на следниот начин: На секоја логичка страница одговара една физичка а коресподенцијата помеѓу нив се чува во табела на страници. Тоа овозможува да континуираниот логички простор на процесот се расфрли насекаде по меморијата на ниво на грануларност од една страница.

1. **Како се реализира хардверска поддршка на страничење?**

Страничењето реализирано со помош на структурите во работната меморија или уште полошо во комбинација на дискот и меморијата е многу споро и неефикасно. Доколку страниците се мали тие ќе бидат многу на број а со тоа и ќе биде поголема табелата на рамки. Исто така голем број на процеси повлекува и голем број на табели на страници за процеси. Секој ОС има сопствена шема за реализација на табела за страници. Покрај главната табела голем број на ОС имаат посебна табела за секој процес. Идеално место за сместување на страниците се регистрите на процесорот како најбрзи компоненти, но бројот на регистри не е доволен. Затоа табелите на страниците кои се релативно големи се чуваат во работната меморија при што обично постои посебен пар на регистри

PTBR( page-table base register)- покажувач на табели на станици

PRLR(page- table length register)- покажувач на големина на табелите на страниците.

1. **Што претставува сегментација и како функционира?**

Сегментација е метод за управување со меморијата кој поддржува логички кориснички поглед врз меморијата. Логичкиот адресен простор се состои од колекција на сегменти, а секој сегмент има единствено име и содржина  
Логичката адреса се состои од два дела:  
\*име на сегментот(наместо името на сегментот обично се задава број кој претставува идентификатор на сегментот)  
\*поместување внатре во сегментот.  
При сегментација се јавува екстерна фрагментација. Слободната меморија не може да се искористи за сместување на сегмент доколку не постои доволно голема празнина,без оглед на вкупната количина на слободна меморија.  
Проблемот на екстерната фрагментација, која зависи од големината на сегментот и распределбата на надојдените процеси, може да се намали со збивање на меморијата.

1. **Што претставува сегментација со страничење?**

Кај процесите кои имаат вградено поддршка и за сегментација и за страничење(од Intel 80386), се овозможува примена на комбинирани методи на дисконтинурана алокација на процесите па така процесите може да се делат на физички дисконтинуирани логички целини. Страничењето е реализирано во две нивоа:  
\*надворешна табела наречена каталог на страници  
\*внатрешна табела - табела на страница.

1. **Од што се состои виртуелната адреса кај сегментацијата со страничење?**

Виртуелната адреса се состои од:

**\*Број на сегментот:** се користи за индексирање на табелата на сегменти чиј влез ја дава почетната адреса на табелата на страници за тој сегмент.

**\*Број на страници:** се користи за индексирање на табелата на страници за да се добие соодветен број на рамка.

**\*Поместување (офсет):** се користи за да се лоцира зборот внатре во рамката.

1. **Што опфаќа сигурноста и заштитата на ОС?**

Проблемот на заштита на ОС се сведува на контрола на пристапот кон објектите на оперативниот систем . За разлика од заштитата која се однесува на контрола на пристапите и внатрешноста на системите, сигурноста опфаќа повеќе аспекти и вклучува интеракција меѓу системите и надворешниот свет. Пробивањето т.е. нарушувањето на сигурноста на системите може да биде случајно или намерно и секако дека е многу полесно да се заштити системот од случајни напади (погрешно користење). Сигурноста опфаќа заштита на системите од:

-Неовластен пристап кон податоци и ресурси

-Злонамерна измена на податоци

-Злонамерно уништување на податоци и

-Спречување на легитимно користење на системот

1. **Што претставува домен на заштита?**

Проблемот на заштита се сведува на контрола на пристапот кон објектите на оперативниот систем: на објектите можат да им пристапат само оние корисници кои се овластени и над објектот можат да се извршат само дозволени операции. Секој домен дефинира множество на објекти и сите операции кои можат да се извршат на тоа множество. Можноста да се изврши операција над објектот се нарекува **право на пристап (access right**). **Доменот претставува колекција на права на пристап на дефинирани парови** (име на објектот, множество на права).

1. **Што претставува матрица на право на пристап и како се имплементира?**

Заштитата може да се прикаже како матрица на пристап во која редовите претставуваат домени а колоните претставуваат објекти. Елементот на матрицата (i , j) претставува збир на операции кои процесот од доменот Di може да ги изврши над објектот Oj. Матрицата регулира како процесите кои припаѓаат на различни домени пристапуваат кон објекти во системот. Меѓутоа во вака дефинирана матрица процесите во одредени ситуации можат да преминат од еден домен во друг домен на заштита и со тоа да остварат поголемо право над објектот. Заради тоа се воведува одредена контрола за преминување на процесите од еден во друг домен на заштита. Матрицата на пристап може да се имплементира на 4 начини во зависност од збирот на домени/објекти.

-Глобална табела- се состои од збир на уредени тројки (домен, објект, збир на права) предност му е што има централизирана заштита на системот а недостаток е големината на табелите.

-Листа за контрола на пристап над објектите- се формира посебна листа за секој објект на системот и одговара на една колона на матрицата на пристап. Листата ја сочинуваат збир уредени парови (Домен, множество на права )

-Листа на множества на домените- оваа листа се формира за секој од домените и одговара на една вид на матрица за право на пристап ја сочинуваат множество на уредени парови (објект, право на пристап)

-Механизам на клучеви- претставува компромис на претходните два начини. На секој објект му се доделува листа на брави а на секој домен листа на клучеви. Клучевите и бравите се едноставни низи на битови. Овој механизам е флексибилен и ефикасен.

1. **Кои се аспекти на безбедност како се врши идентификација на корисник?**

**Аспекти на безбедност се:**

**-Физичко ниво-** компјутерите и мрежната опрема неопходно е да бидат физички обезбедени од оган, влага, кражба и сл.

**-Човечки фактор-** како системски администратори потребно е да се изберат луѓе од доверба.

-**Мрежно ниво-** оддалечен пристап кон ресурсите за легитимните корисници на системот, заштита на ресурси од неовластен пристап итн.

-**Ниво на ОС-** ОС мора да се заштитат самите себе и системот во целина од случајни напади или намерни оштетувања.

**\*Идентификација на корисник**

Еден од главните проблеми со безбедноста на ОС е автентичноста на корисниците - ОС на некој начин мора да знае дали се работи за правиот овластен корисник или не. Поради тоа секој корисник пред користење на системот мора да се идентификува. Генерално идентификацијата може да се оствари на три начини:

-**специјален хардвер како што е клуч или ID картичка**

**-внесување на доверливи информации како што се лозинката**

**-биолошки атрибути на корисникот (отпечаток на прстите, потпис, снимање на очна рожница итн.)**

1. **Да се објасни некој од методите на напад!**

**\*Напад на системот-** системот може да биде нападната на повеќе начини најчесто користени видови на напад се неовластениот пристап и експлоатација на познатите слабости на програмата. Најчести користени методи на експлоатација се одбивање услуга, лажирање и нушкање. Администраторот може да ја заштити мрежата од вакви напади со користење на мрежни бариери.

**\*Одбивање на услуга-** DoS предизвикува стопирање на работата на сервисот или програмата што на другите им оневозможува да работата со нив. Овој напад може да се изврши преку праќање на голем број на SYN пакети TCP

Заштитата може да се воспостави преку контролирање на бројот на SYN пакети во единица време.

**\*Лажирање-** при овој начин на напад напаѓачот ги прати IP адресите во IP пакетите и се претставува како друг компјутер. Бидејќи DNS не го проверува изворот на информацијата т.е. од кае таа доаѓа напаѓачот може да изврши напад на DNS сервисот давајќи му погрешни информации. Најдобра заштта од овој вид на напад е спречување на рутирањето на датаграми со неиспрaвни изворни адреси.

**\*Smurf-** Напаѓачот прек интернет им праќа пораки на сите членови на мрежата **ICMP ECHO- REQUEST** пакети и како одредишна IP адреса ја става адресата на жртвата . Жртвата прима голем број на **ECHO-REPLAY** пакети и најдобар начин на заштита е да се отфрлат ping пакетите кои се испратени од сите.

**\*Нушкање-** Метод во кој со специјална програма наречена нушкач се претставува TCP/IP пакети кои поминуваат низ одреден компјутер и по потреба се прегледува нивната содржина. Доколку во мрежата циркулираат нешифрирани податоци нушкачот лесно може да дојде до доверливи информации.

**\*Смртоносен ping –** Праќање на голем број на ICMP REQUEST пакети може да биде фатално за некои ОС. Заштитата се сведува на прифаќање на ограничен број на ICMP REQUEST пакети во единица време или во краен случај забрана за прием на ICMP REQUEST пакети.

**\*Скенирање на приклучоци-** служи за идентификација на отворените приклучоци на жртвата- напаѓачот испраќа SYN или FIN пакети кон опсегот и очекува RST пакети за неактивните приклучоци. Заштитата се постигнува со прифаќање на ограничен број на SYN пакети во единица време.

1. **Кои се програмски закани и да се објасни како функционира преполнување на стек и бафер?**

Програмски закани се:

\***Тројански коњи-** тие се нелегален сегмент на код подметнат во кодот на програмата со цел да се промени функцијата или однесувањето на оригиналната програма. Тоа се програми кои се преправат дека се корисни софтверски алатки,но во реалноста кога ќе се стартува во позадина инсталира опасен и злонамерен софтвер. Тројанците се способни да ја намалат сигурноста на нашиот компјутер да дозволат на други целосен пристап до сите датотеки на компјутерот и сл. Вообичаен тројански коњ се состои од две програми Client и Server. Има неколку начини да се инфицираме со тројански коњ:

-Ако спуштиме инфициран емаил

-Ако спуштиме инфицирана датотека од интернет

-Ако земеме инфицирана датотека преку дискета итн.

**\*Замка-** Авторот на програмата може акајќи или ненамерно да остави празно место во својот код. Оној кој ги знае тие места може да подметне свој код и со тоа да се оствари добивка.

\*\***Преполнување на стек или бафер-** Најчесто упатуван напад од мрежа пи обид за неовластен пристап кон системот. Овој тип на напад можат а го користат и овластени корисници со цел да го измамат системот и да остварат поголеми права од оние кои ги имаат. По правило напаѓачот ја користи грешката во програмата т.е. недоволно контролирање на раздвојувањето на стекот податоците и кодот. Тогаш напаѓачот испраќа повеќе влезни податоци отколку што програмата очекува и ја испробува ранливоста на програмата на следниот начин:

-преполнување на влезното поле аргументите на командна линија или влезниот бафер се додека не дојде до стекот

-преку важечка адреса во стекот се запишува адресата од својот код која се сместува во стекот во следниот чекор

-полнење на дел на стекот со својот код.

Во случај на успешен напад наместо недоволно заштитена програма ќе се изврши илегалниот код вметнат преку полнење на баферот.

1. **Кои се системски закани?**

**\*Вируси-** Вирус претставува вид на системска закана која може да изврши различни деструктивни активности како што се бришење датотеки, цели дискови и сл. Компјутерскиот вирус е програма што може да се постави на даден компјутер и подоцна да се реплицира самиот и да се рашири од еден компјутер до друг. Кога вирусот ќе влезе во компјутерот тој може да биде активен во позадина да испраќа пораки на екранот да свири најразлични звуци да уништува податоци па дури и да го расипе компјутерот. Вирусот се активира секогаш кога компјутерот ќе изведе инфицирана програма т.е. програма што содржи вирус. Додека е во меморијата тој може да предизвика голема штета како што е реплицирање преку компјутерот а ако компјутерот е приклучен на мрежа тогаш може да ги инфицира сите мрежни компјутери. Токму поради овие причини потреба е зголемена заштита од вируси. Најдобра заштита е често освежување на листата на анти-вирусни дефиниции во анти-вирусната програма.

**\*Компјутерски црви-** Компјутерски црв е процес кој користи механизам за размножување со цел да ги деградира системските перформанси. Во мрежното окружување црвите се многу опасни бидејќи можат да се прошират и да ја блокираат целата мрежа. Црвите исто како и вирусите се реплицираат самите себе но разликата е во тоа што црвите постојат како засебна единка т.е. тие не се закачуваат себеси на некоја датотека или програма. Црвите можат да се шират самите преку мрежа од еден компјутер на друг. За откривање и заштита од црвите се користи анти-вирусна програма.

**\*Морисов црв-** Овој црв се состои од главна програма која се обидува да ги открие лозинките на корисникот и јадица. Заразениот систем со техниката преполнување на бафер или стек се обидува да вметне јадица на друг систем во мрежата користејќи ги слабостите на програмите rsh, finger, sendmail . Во случај кога црвот успева да ја пробие јадицата таа поднесува барање до заразениот компјутер од кој бара копија на црвот. Црвот се копира и понатаму двете машини продолжуваат да ги шират црвот низ мрежата. Морисовиот црв во 1998 година направил голема штета на UNIX системите.

1. **Да се набројат неколку техники за зголемување на сигурноста на системот!**

Еден од начините да се зголеми општата безбедност на системите се сведува на периодична проверка на можни безбедносни пропусти во системот.

Потребно е да се провери дали постојат:

-кратки лозинки или лозинки кои лесно се погодуваат

-неочекувани процеси кои долго се извршуваат

-опасни програми со доменски бит

-опасни влезови во програмската патека

-неавторизирани програми во системскиот директориум

-несоодветна заштита за директориумот

-несоодветна заштита за системскиот директориум

-промени во контролните збирови на системските програми.

1. **Кои се функциите на влезно – излезните потсистеми?**

**--**Ги управува влезно – излезните уреди и операциите кои тие уреди ги извршуваат и ги контролираат.

**--**Обезбедува што поедноставен интерфејс кон корисникот и остатокот од системот.

1. **Што е и што се постигнува со независноста на уредите?**

Постојат два аспекти на независност на уредите:

-**Независноста на програмата од конкретниот модел на уред кој се користи**- различните модели на уредите од истиот од аспект на програмата мора еднакво да се набљудува.

**-Независност на програмата од конкретниот вид кој се користи**-

1. **Што претставува независност на знаковниот код?**

Очигледно непожелно е да се пишуваат програми кои бараат од корисникот детално познавање на знаковниот код користен во конкретните уреди. Влезно – излезниот потсистем мора да преземе грижа за препознавање на конкретниот знаковен код и за претставување на податоците на корисникот во стандарден облик. Последица на независноста на знаковниот код е постоење на внатрешен код т.е.:

-униформни претстави на сите знаци во самиот уред

-механизми за преведување на знаковниот код кои обично се реализирани во форма на транслациони табели или во форма на помали програми.

1. **Како се постигнува ефикасност?**

Секој уред е многу побавен од работната меморија и самиот процесор што значи дека влезно – излезните операции се тесно грло на компјутерскиот систем и дека е пожелно да се извршат што е можно поефикасно. На ниво на ОС за да се изврши зголемување на ефикасноста на извршувањето на влезно – излезните операции ВИ потсистем мора да обезбеди конкурентност во работата со урдите у различни механизми на кеширање, баферирање и користење спулери.

1. **Униформен интерфејс кон апликациите и корисникот!**

Пожелно е да се овозможи униформно ракување со уредите за да им се

овозможи поедноставна работа на корисниците. Заради разновидноста на уредите оваа цел е тешко да се постигне. Можно решение на овој проблем кој се користи UNIX и Linux системите.

1. **Класификација на влезно – излезни уреди?**

Уредите се класифицираат според следните карактеристики:

\***Намена на уредите**

**-**уреди за долготрајно сместување на податоци (дисковите и лентите)

**-**уреди за пренос на податоци (мрежни картички и модеми)

**-**уреди кои обезбедуваат интерфејс кон корисникот(монитор,маус,тастатура)

**\*Насока на пренос**

**-**влезни

**-**излезни

**-**влезно – излезни

**\*Единечна количина на пренесени податоци**

**-**уреди кои работат со блокови (диск, трака)

**-**уреди кои работата со знаци (терминал, маус, печатач)

**-**мрежни уреди

**\*Метод на пристап**

-уреди со секвенцијален пристап (модем и магнетната лента)

-уреди со директен или случаен пристап (дисковите CD-ROM и DVD уреди)

**\*Деливост на уредите**

**-**Деливи уреди (дискови)

**-**Неделиви (тастатура и маус)

**\*Брзина на уредите**

**-**брзи

**-**спори

**\*Можност на записот**

**-**оние кои служат за читање и пишување(дисковите)

**-**оние кои служат само за читање (CD-ROM уредите)

**-**оние кои служат само за пишување (графичка картица)

1. **Објасни како работи часовникот и тајмерот !**

**Поголемиот број на компјутери имаат хардверски часовници и тајмери кои обезбедуваат три основни функции:**

**-Прикажување на тековното време**

**-прикажување на изминатото време**

**-тајмерски прекинувач за операцијата Х во Т**

Часовникот и тајмерите претставуваат специјална класа на уреди така да им се пристапува со помош на специфични системски повици. Програмабилните интервалски тајмери мерат изминато време или обезбедуваат прекинување. Тие се програмираат така што после одредено време генерираат прекинувачки сигнал. Часовник и тајмер се користат за реализација на функциите на предпразнење. Тајмерите кои го мерат изминатото време или после одредено време генерираа прекинувачки сигнал се користат за реализација на техника на предпразнење при распоредување на процесорите.

Претпразнењето функционира на следниот начин

-тајмерот генерира прекинувачки сигнал

-се прекинува извршувањето на тековниот процес,а потоа

-се повикува диспечерот да

-врз основа на алгоритмот за распоредување на процесорот го одбере процесот од редот на процесорот

-и му го предаде процесорот на користење

1. **Како работи контролерот?**

**Контролерот работи на три начини**

**-**како приклучок (Port)

**-**како магистрала (Bus)

**-**како уред (device)

1. **Како може да се пристапи до регистрите со оглед на меморијата и ВИ просторот?**

До регистрите на компјутерот може да се пристапи на два начини во зависност од тоа дали се разделени меморискиот и влезно – излезниот простор. Доколку влезно – излезниот простор може да се третира како меморија трансферите може да се извршат со обични инструкции за работата со меморијата (мемориски мапирани В/И операции) Типичен пример е меморијата на графичките картички која служи за сместување на слики.

Типичен влезно – излезен приклучок се состои и 4 вида на регистри:

1. **Од што се состои ВИ приклучок?**

\***Контролен регистер**

**-**Служи за поставување на работата на уредот како што се брзината на приклучокот и типот на комуникација. Во овој регистер процесорот исклучиво запишува податоци.

**\*Статусен регистер**

**-**Овој регистер го опишува статусот на командите што ги извршува т.р. дали командата е извршена дали е спремен податокот дали настанала грешка итн. Процесорот од статусниот регистер исклучиво ги чита податоците.

**\*Регистер на податоци о влезен режим**

**-**Влезниот регистер служи за читање на податоци од влезните уреди. Процесорот од влезниот регистер исклучиво ги чита податоците

**\*Регистер податоци за излезен режим**

-Излезниот регистер служи за запишување податоци на излезниот уред . Од овој регистер процесорот исклучиво чита информации.

1. **Техника на повикување на уреди!**

Техниката на повикување на уреди- повикувањето на уредите се извршува со јамка:

-**компјутерот го повторува читањето на статусниот регистер и**

**-го анализира busy битот** се додека вредноста на битот не стане 0

**Основен** недостаток на техниката на прозивање е трошење на процесорското време на јазолот за прозивање.

1. **Механизмот на прекин!**

Недостатокот може да се отстрани со воведување на хардверски механизам кој овозможува уредот да му означи на процесорот кога е командата завршен. Овој механизам се нарекува прекин. Прекинувачкиот систем функционира на следниот начин:

-кога уредот ќе ја заврши операцијата контролерот му пушта прекинувачки сигнал на процесорот преку прекинувачка линија

-процесорот ќе ја заврши тековната инструкција

-после се прекинува секвенцијално извршување на програмот, како би се обработил прекинот

-после обработката на прекинот процесорот продолжува со извршувањето на програмата.

1. **Приоритетен прекинувачки контролер (PIC) !**

-можност на одложување на обработката на прекинот додека процесот е во критична состојба

-брза и ефикасна техника на одредување на уредот кој што го поставил прекинот

-овозможено вгнездување на прекинот

-повеќе - слоен прекинувачки систем кои ќе разликуваат приоритетот на прекинувачките сигнали и по приоритет ќе ги решаваат прекините .

Сите овие функции ги реализира хардверски уред кој се нарекува Приоритетенпрекинувачкиконтролер **(**PIC**).**

1. **Да се објаснат двете линии за пренос на прекинувачки сигнал!**

**Приоритетен прекинувачки контролер (PIC)-** повеќето процесори наместо по еден имаат по две посебни линии за прекинувачки сигнал

**\*Немаскирна линија**

**-**Прекинувачкиот сигнал испратен по немаскирната линија секогаш може да го прекине извршувањето на тековниот процес. Немаскираната линија се користи за праќање на прекинувачки сигнал заради критичните хардверски грешки во меморијата.

**\*Маскирна линија**

- Прекинувачкиот сигнал испратен по маскирната линија не го прекинува извршувањето на тековниот процес се додека пример процесот се наоѓа во критична секција. Маскирната линија се користи за праќање на прекинувачки сигнали кои потекнуваат од нормални операции и стандардните грешки кои се јавуваат на ВИ уреди.

1. **DMA!**

DMA се користи за да се избегне програмирањето на ВИ за трансфер на големи количини на податоци.

**DMA контролер -**  за да може да се направи брз пренос на податоци кај уредите како што е дискот (кој има огромно количество податоци) на релација контролер – работна меморија и обратно

**-**Сигнали за управување со трансферот од контролерот

**-**сигнали за управување со мемориските циклуси

**-**dma мора да се програмира

**-**компјутерот содржи повеќе DMA канали и секој има свој ВИ контролер

**Со DMA се бајпасира (одбегнува) процесорот при трансфер на ВИ уред и меморија.**

1. **Работење на интерфејс!**

Интерфејс е стандардно множество на функции преку кое се обраќаме на истата класа на влезно – излезни уреди. Разликите кои постојат помеѓу уреите од истата клас скриени се во специјални модули на јадрото од ОС кои се нарекуваат управувачки програми или драјвери. Драјверите постојат за секој уред но на корисникот им се претставуваат како еден универзален интерфејс. Над сите драјвери се наоѓа слој на влезно – излезен потсистем на јадрото кој практично е независен од хардверот- сите детали за конкретен драјвер се препуштени на драјверите.

1. **Услуги што ги обезбедува ВИ!**

**-**управување со имињата на датотеките и урдите

**-**контрола на пристапот во датотеките и уредите

**-**контрола на операциите

**-**доделување на уредите на процесите на користење

**-**распоредување на влезно – излезните операции

**-**баферување, кеширање, спулер

**-**контрола на статусот на уредите

**-**конфигурирање и иницијализирање на драјверот

1. **Распоредување на ВИ операции!**

Распоредување на влезно – излезни операции- обезбедување на добар поредок на извршување на влезно излезните операции а со цел постигнување на правилни перформанси.

-Поредокот по кој се пристигнати во системот, барањата за извршување на влезно излезните операции по правило е строго стохастички и скоро никогаш не е оптимален. Распоредувањето на операциите се потпира на редот на чекање за секој уред посебно.

1. **Управување со грешки!**

Уредите и влезно – излезните трансфери можат да откажат од многу причини и тоа привремени или трајни.

Во случај на привремени грешки оперативниот систем може самиот да се опорави така што ќе ја повтори операцијата (try во случај на дискот, resend во случај на мрежата)

Доколку се случи трајна грешка нема ваква помош, а ОС единствено може да го искористи корисникот дека се случила грешка. Известување на корисникот обезбедува системски повици кои го враќаат статусот на извршување на операцијата т.е. извештајот за успешност. Кога ќе се случи грешка системскиот повик мора прецизно да го опише нејзиниот вид и локација.

1. **Спулер!**

**Спулер е бафер кој привремено ги чува излезните податоци наменети за некој неделив уред.** Истовремено спулерот овозможува пристап до неделивите уреди на следниов начин:

-Процесите ги запишуваат податоците наменети за уредот на дикс а

-ОС управува о спулерот на тој начин што ги опслужува барањата едно подруго.

Користењето на суплерот има свои предности

-процесот релативно брзо го поставува своето барање во бафер,а после тоа е слободен понатаму да ги извршува своите активности

-неделивите уреди се користат како привидно деливи што овозможува на поголем број процеси истовремено да го користат уредот.

Кога не би се користел суплер процесите би морале да чекаат уредот да се ослободи за да можат да го користат а потоа повторно и се блокирале и би чекале уредот да ја заврши операцијата која тие ја иницирале.

1. **Кеширање**

Кеш(cache) претставува област на брза системска меморија која ја чува копијата на податоците, најчесто на дискот. Пристапот на податоците во кешот е значително побрз од пристапот кон податоците на влезно-излезните уреди. Кеширање(cache) претставува техника на копирање на делови од дискот во кеш меморија,со што значително се подобруваат перформансите на В/И системите од дискот.

1. **Бафер!**

Бафер е дел од меморијата која функционира на принцип на

производител - потрошувач и служи за привремени податоци при пренос на податоците помеѓу два уреди или помеѓу уред и апликација.

Цели:

-Израмнување на различните брзини помеѓу потрошувачот и производителот

-Прилагодување на различни количини на трансфер на податоци

-Одржување на семантиката на копирање

1. **Разлики помеѓу кеш и бафер!**

Баферот моментално ги чува актуелните податоци а

-кешот чува било која копија од дискот.

-Ист мемориски простор може да се користи и за баферување и за кеширање.

По правило кеширањето е многу значајно и се реализира во повеќе ниво при што се користат различни техники за пополнување на податоци во кешот.

1. **Прилагодување на различни количини на трансфер на податоци?**

Прилагодување на различни количини на трансфер на податоци- Баферите многу често се користат за мрежни околини каде големите пораки се префрлуваат во делови поделени на помали пакети. Пакетите се примаат во бафер по недетермонистички редослед (кое зависи од појавата не грешки во мрежата) после тоа од баферот се реконструира почетна порака.

1. **Семантика на копирање!**

Корисничкиот процес издава системски повик за запишување односно испраќање на податоци на некој уред. Доколку повикот е неблкирачки или асинхрон процесот добива можност за време на испраќањето да ги измени податоците кои што сака да ги испрати на уредот. Тоа значи дека процесот може да запиши нешто друго, а не она што е зададено иницијално. За да се избегни тоа т.е. да се зачува семантиката на копирање ОС во своето јадо го дефинира баферите во кои го копира корисничкиот бафер па потоа ги предава под контрола на корисничкиот процес.

1. **Разлика помеѓу блокирачки и неблокирачки операции!**

**Блокирачки-** процесот е суспендиран се додека не се комплетира БИ операција

**-**Лесни за користење и разбирање

**-**Недоволни за некои потреби

**Неблокирачки-** БИ повик враќа што е овозможено

**-**кориснички интерфејс, копирање на податоци (бафериран ВИ)

**-**применето преку повеќе нитки

-Враќа брз одговор со број на бајти кои се прочитани или запишани

* **Кај континуирано доделување на меморија,бит мапите претставуваат**

Динамична алокација на мемориите

* **Под кои услови се појавува застој?**
* **За јадрото да ја реализира својата функција неопходно е самиот хардвер да исполни одредени предуслови,односно да поседува одредени компоненти.Кои се тие компоненти?**
* **Кои се програмски закани за системот?**

**-**Тројански коњи , Замка и Преполнување на стек или бафер**.**

* **Опишете го проширениот век на процесорот:**
* **Под кои услови се појавува застој:**

1. Линеарно чекање
2. При користење на ресурсите се почитува принципот на меѓусебно исклучуавње
3. Кружно чекање
4. Доделување на ресурсите се обавува без претпразнење-на процесот не може да му се одземе ресурс
5. Процесот здружува еден ресурс бидејќи му е потребен и чека за ресурс кој го користи некој друг процес
6. Доделување на ресурсите се обавува со претпразнење- на процесот не може да му одземе ресурс
7. При користењето на ресурсите не се почитува принципот на меѓусебно исклучување

* **Во што е разликата помеѓу распоредување без предпразнење и распоредување со предпразнење?**
* **За да јадрото ја реализира својата функција неопходно е самиот хардвер да исполни одредените предуслови, односно да поседува одредени компоненти. Кои се тие компоненти?**
* **Една од функциите на планерот на работи е?**

1. Ги дели работите на процеси
2. Извршува суспендирање на процесот
3. Доделува процесор на процесите кои што се наоѓаат во процесорскиот ред

* **Кои се предностите од употребата на програми со повеќе нишки?**

1. Спречување на програмата да велзе во deadlock

* **Контролерот работи на три начини. Графичката картичка е:**
* **Што претставува MMU**
* **Опишете го проширениот животен ве на процесот?**
* **Софтвер со кој се пресретнуваат TCP/IP пакети кои поминуваат низ одреден компјутер и по потреба се прегледува нивната содржина претставува:**

1. Sniffing
2. Смтроносен ping
3. Smurf
4. dos

* **Што е контролен блок на процес?**

1. Дел од меморијата односно мемориска структура со основни информации за процесот кој што OC го користи за управување со тој процес
2. Дел од оперативниот систем кој овозможува контрола, извршување и управување со процесите

* **Кои се дистрибуирани OC? (може да изберете повеќе опции)**

1. Систем во кој секој компјутер има свој оперативен систем
2. Систем во кој меѓусебно можи да се разменуваат податоци со помош на одредени протоколи
3. Систем во кој компјутерите содржат релативно висок степен на автономија
4. Систем кој освен споделување и миграција на датотеки овозможува и споделување на процеси т.е програми

* **Дали екстерната фрагментација е карактеристика за континуирана или дисконтинуирана алокација на меморијата?**
* **Контролниот регистер кај типичен влезно-излезен приклучок**

1. Служи за поставување режим на работата на уредот
2. Служи за да покажува дали командата е извршена
3. Служи за запишување на податоци на излениот уред

* **Во што е разликата помеѓу распоредување без предпразнење и распоредување со препразнење?**

1. Во случај на распоредување со претпразнење, процесорот може да му се одземе на процесот кој не ги завршил своите активности. Во случај на распоредување без препазнење, тоа не е можно
2. Во случај на распоредување со претпразнење, процесорот може да му се доделува на процес со највисок приоритет. Во случај на распоредување без претпразнење тоа не е можно
3. Во случај на распоредување без претпразнење, процесорот може да му се одземе на процесот кој не ги завршил своите активности. Во случај на распоредување со претпразнење, тоа не е можно

* **Кои се програмски закани за системот? (изберете повеќе опции)**

1. Лажирање (spoofing) на DNS
2. Тројански коњи
3. Замка (trap door)
4. Преполнување на стекот или баферот
5. Smurf

* **Што е контролен блок на процес?**
* **Кои се дистрибуирани OC?**

1. Систем во кој секој компјутер има свој оперативен систем
2. Систем во кој меѓусебно можи да се разменуваат податоци со помош на одредени протоколи
3. Систем во кој компјутерите содржат релативно висок степен на автономија
4. Систем кој освен споделување и миграција на датотеки овозможува и споделување на процеси т.е програми

* **Кеш меморијата се користи за**

1. Област на брза системска меморија која ја чува копијата на податоците
2. Да чува било која копија од дискот
3. Да ги чува актуелните податоци
4. Да се избегне програмирањето на ВИ за трансфер на големи количини на податоци

* **Кои се предностите од употреба на програми со повеќе нишки?**

1. Спречување програмата да влези во deadlock
2. Подобро искористување на повеќепроцесорската архитектура
3. Намалување на времето на одзив
4. Поефикасно делење на ресурсите и економичност

* **Кај континуирано доделување на меморијата, бит мапите претставуваат**

1. Динамичка алокација на меморијата
2. Статичка алокација на меморијата
3. Статичко поврзување на меморијата
4. Динамичко поврзување на меморијата

* **Контролерот работи на три начини. Графичката картичка е:**
* **Контролниот регистар кај типичен влезно-излезен приклучок**

1. Служи за поставување режим на работата на уредот
2. Служи за да покажува дали командата е извршена
3. Служи за запишување на податоците на излезниот уред

* **Кај континуирано доделување на меморијата, системот на здружени парови- другари (buddy system) претставува**

1. Динамичка алокација на меморијата
2. Динамичко поврзување на меморијата
3. Статичка алокација на меморијата
4. Статичко поврзување на меморијата

* **Дали екстерната фрагментација е карактеристика за:**

1. Континуирана алокација на меморијата
2. Дисконтинуирана алокација на меморијата
3. Двата вида алокација на меморијата

* **Што претставува MMU**
* **Што е контролен блок на процес**

1. Дел од меморијата односно мемориска структура со основни информации за процесот кој што OC го користи за управување со тој процес
2. Дел од опреативниот систем кој овозможува контрола, извшување и управување со процесите

* **Кои се дистрибуирани OC? (може да изберете повеќе опции)**

1. Систем во кој меѓусебно можи да се разменуваат податоци со помош на одредени протоколи
2. Систем во кој секој компјутер има свој оперативен систем
3. Систем во кој компјутерите содржат релативно висок степен на автономија
4. Систем кој освен споделување и миграција на датотеки овозможува и споделување на процеси т.е програми

* **Во што е разликата помеѓу распоредување без предпразнење и распоредување со предпразнење?**

1. Во случај на распоредување со предпразнење, процесорот може да му се одземе на процесот кој не ги завршил своите активности. Во случај на распоредување без претпразнење, тоа не е можно
2. Во случај на распоредување со претпразнење, процесорот може да му се доделува на процес со највисок приоритет. Во случај на распоредување без претпразнење, тоа не е можно
3. Во случај на распоредување со препразнење, процесорот може да му се одземе на процесот кој не ги завршил своите активности. Во случај на распоредување без претпразнење, тоа не е можно.

* **Стопирање на работата на сервисот или програмата така што на другите корисници им се оневозможува да работат со нив претставува:**

1. Смртоносен ping
2. Dos
3. Smurf
4. Sniffing

* **Дали екстерната фрагментација е карактеристика за**

1. Континуирана алокација на меморијата
2. Дисконтинуирана алокација на меморијата
3. Двата вида на алокација на меморијата

* **Софтверот со кој се претставуваат TCP/IP кои поминуваат низ одреден компјутер и по потреба се прегледува нивната содржина претставува:**

1. Sniffing
2. Smurf
3. Смртоносен ping
4. Dos

* **Кои се предностите од употреба на програми со повеќе нишки?**

1. Поефикасно делење на ресурсите и економичност...
2. Спречување програмата да влези во deadlock
3. Подобро искористување на повеќепроцесорската архитектура
4. Намалување на времето на одзив

* **Кеш меморијата се користи за**

1. Да се избегне програмирањето на ВИ за трансфер на големи количини на податоци
2. Област на брза системска меморија која ја чува копијата на податоците
3. Да ги чува актуелните податоци
4. Да чува било која копија од дискот

* **Бришење на сите фајлови од фолдерот c:\sistemski**

Echo 1. Промена на наслов

Echo 2. Печатење со лооп

Echo 3. Бришење на фајлови

* **За јадрото да ја реализира својата функција неопходно е самиот хардвер да исполни одредените предуслови, односно да поседува одредени компоненти. Кои се тие компоненти?**
* **Во што е разликата помеѓу распоредување без предпразнење и распоредување со предпразнење**

1. Во случај на распоредување со претпразнење, процеосорот може да му се одземе на процесот кој не ги завршил своите активности. Во случај на распоредување без претпразнење, тоа не е можно
2. Во случај на распоредување со претпразнење, процесорот може да му се доделува на процес со највисок приоритет. Во случај на распоредување без претпразнење, тоа не е можно
3. Во случај на распоредување без претпразнење, процесорот може да му се одземе на процесот кој не ги завршил своите активности. Во случај на распоредување без претпразнење, тоа не е можно

* **Контролниот регистер кај типичен влезно-излезен приклучок**

1. **Нема одговори**

* **Нема прашање**

1. Пуштање на пораки
2. Користење делливи мемории
3. Изменувани цевки

* **На кои се начини може да се изврши идентификација на корисникот на OC?**
* **DMA се користи за**

1. Да се избегне програмирањето на ВИ за трансфер на големи количини на податоци
2. Да чува било која копија од дискот
3. Област на брза системска меморија која ја чува копијата на податоците
4. Да ги чува актуелните податоци

* **Дали е можно системо да се најде во состојва на застој до колку на него постои само еден процес?**

1. Не е можно. Ова е директна последица на условите на задржување на ресурсите и чекање на други ресурси
2. Можно е. Ова е директна последица на условите на задржување на ресурсите и чекање на други ресурси

* **Што претставува интерна фрагментација?**

1. Просторот од меморијата што останува неискористен меѓу партициите
2. Просторот од меморијата што останува неискористен во партициите
3. Просторот од меморијата што останува неискористен меѓу партициите на почетокот на доделувањето

* **Статусниот регистер кај типичен влезно-излезен приклучок**

1. Служи за поставување режим на работата на уредот
2. Служи за запишување на податоците на излезниот уред
3. Служи за да покажува дали командата е извршена

* **Една од функциите на средниот распоредувач е да**

1. Доделува процесор на процесите кои што се наоѓаат во процесорскиот ред
2. Ги дели работите на процеси
3. Извршува суспендирање на процесот

* **Под кои услови се појавува застој? (изберете повеќе од 3 услови)**

1. Кружно чекање
2. Линеарно чекање
3. Доделување на ресурсите се обавува без претпразнење – на процесот не може да му се одземе ресурс
4. При користењето на ресурсите се почитува принципот на меѓусебно исклучување
5. При користењето на ресурсите не се почитува принципот на меѓусебно исклучување
6. Процесот задржува еден ресурс бидејќи му е потребен и чека за ресурс кој го користи некој друг процес
7. Доделување на ресурсите се обавува со претпразнење – на процесот не може да му одземе ресурс

* **Контролерот работи на три начини. Серискиот контролер е**
* **Што претставува дел од јадрото на OC?**

1. Планер за работи од ниско ниво
2. Семафор
3. Планер за работи од средно ниво
4. Именувани цевки
5. Планер за работи од висок ниво
6. Рутини за обработка на прекини

* **Која е задачата на диспечерот?**

1. Бира процес од редот на чекање
2. Доделува процесор на процесите кои што се наоѓаат во процесорскиот ред
3. Извршува суспендирање на процесот