

Тема 1

- ✓ *Опишете как ОС разделят ресурсите на изчислителната система, дайте примери за основните типове разделяне:
Разделяне на пространството (памети).
Разделяне на времето (процесори, други у-ва).*
- *Опишете с по едно-две изречения работата на следните системни извиквания в стандарта POSIX:
pipe() dup2() fork() exec() wait() waitpid()*

ОС разделят ресурсите на изчислителната система с цел ефективно и справедливо използване на наличните хардуерни и софтуерни компоненти от множество процеси и потребители.

Основните типове разделяне са:

1. Разделяне на пространството:

Тук ресурсите, като оперативната памет (RAM) и дисковото пространство, се разпределят логически между различните процеси или потребители. Пример за това е виртуалната памет – всеки процес работи в собствено виртуално адресно пространство, което чрез техники като страниране или сегментиране се съпоставя с физическата памет. Това позволява на процесите да използват по-голям адресен обхват, отколкото е наличният физически размер на RAM.

2. Разделяне на времето:

При този подход ресурсите, като процесорното време и достъпа до други устройства (например принтери или мрежови адаптери), се разпределят на малки времеви интервали – кванти. Всеки процес получава редовен дял от процесорното време, така че да изглежда, че работи едновременно с другите, въпреки че физически процесорът изпълнява само една задача в даден момент. Този метод осигурява ефективно използване на процесора и намалява времето за реакция на системата.

Тема 2

- ✓ *Опишете разликата между времеделене и многозадачност.*
- *Какви ресурси разделя еднозадачна, еднопотребителска ОС?*
- *Опишете с по едно-две изречения работата на следните системни извиквания в стандарта POSIX:
open() close() read() write() lseek()*

Времеделенето и многозадачността са техники за абстракция, използвани за ефективно разпределение на системните ресурси, като например оперативната памет (RAM).

Многозадачността представлява по-прост механизъм, приложим още в първите операционни системи. При нея всеки процес получава изчислително време последователно, като превключването между процесите може да става в сравнително дълги интервали. Това означава, че не може да се установи в момента кой процес работи – процесите „работят тайно“, а важното е фактивното

използване на изчислителната система.

При времоделенето се цели няколко активни процеса да изглеждат, че работят едновременно, като всеки процес има впечатление, че разполага с целия процесор. Това е по-сложна техника за разделяне, при която времето се разделя на малки кванти, които след това се предоставят последователно на активните процеси.

Тема 3

- ✓ *Дайте кратко определение за: многозадачна ОС, многопотребителска ОС, Времоделене.*
- *Опишете разликата между многопотребителска и многозадачна работа.*
- *Какви качества на ОС характеризират тези две понятия?*
- *Опишете с по едно-две изречения работата на следните системни извиквания в стандарта POSIX:*
open() close() lseek() pipe() dup2()

Многозадачна ОС: Операционната система, която може да изпълнява няколко задачи (процеса) едновременно чрез превключване на процесорното време между тях.

Многопотребителска ОС: Операционна система, която позволява на множество потребители да използват системните ресурси едновременно, като гарантира изолация и контрол на достъпа до ресурсите.

Времоделене: Техника за разпределяне на процесорно време, при която времето се разделя на малки интервали (кванти) и всеки активен процес получава свой дял, така че да изглежда, че работи самостоятелно.

Тема 4

- ✓ *Опишете ситуацията съревнование за ресурси (race condition), дайте пример.*
- *Опишете накратко инструментите за избягване на race condition:*
(а) дефинирайте критична секция, атомарна обработка на race condition.
(б) инструменти от ниско ниво, специфични хардуерни средства.
(в) инструменти от високо ниво, които блокират и събуждат процес.
- *Каква е спецификата на файловете в следните директории в Linux:*
/etc /dev /var /boot /usr/bin /home /usr/lib /var/log

Състезанието за ресурси (race condition) възниква, когато два или повече процеси (или нишки) едновременно достъпват споделен ресурс – например структура от данни в общата памет – и поне един от тях я модифицира без подходяща синхронизация. Това може да доведе до нарушаване на инвариантите (състояния, които трябва да останат непроменени) на структурата, като поради това може да се загубят или повредят данни.

Пример:

Да предположим, че имаме структура от данни, съдържаща брояч, който винаги трябва да бъде положително число. Ако два процеса едновременно се опитат да увеличат стойността на брояча без да използват механизъм за синхронизация, е възможно и двамата да прочетат една и съща стара стойност, да изчислят новата стойност независимо един от друг и след това да запишат резултата. В този случай

едно от увеличението може да бъде загубено, защото промените се базират на една и съща начална стойност.

За да се предотврати race condition, се дефинира критична секция – това е част от кода, където се извършват операции, които временно нарушават нормалното (статично) състояние на структурата. Докато един процес работи в критичната секция, останалите процеси трябва да бъдат блокирани, за да не се опитват да модифицират същия инвариант. Атомарната операция представлява неделима операция, която се изпълнява изцяло или изобщо не се изпълнява – тя гарантира, че промените се извършват без да се прекъсват от други процеси, като по този начин се избягва race condition

Spinlock(инструмент от ниско ниво):

В съвременните паралелно работещи ядрови системи методът с забрана на прекъсванията не е приложим. Вместо това се използва допълнителен бит – наречен lock – който показва дали данните в критичната секция са свободни или се използват. При този подход процесът, който иска достъп до критичната секция, извиква процедурата spin_lock, която чрез операцията test-and-set проверява и "заклучва" ресурса. След приключване на критичната секция ресурсът се освобождава чрез spin_unlock.

Семафор(инструменти от високо ниво)

Семафор е синхронизиращ механизъм от високо ниво. Дефиниран от Дейкстра около 1965. Използва механизми за смяна на контекста.

Тема 5

- Хардуерни инструменти за защита (lock) на ресурс:
- (a) enable/disable interrupt
- (b) test and set
- (c) atomic swap
- ✓ Опишете инструмента spinlock, неговите предимства и недостатъци.
- Каква е спецификата на файловете в следните директории в Linux:
/etc /dev /var /proc /bin /home /usr/doc

Метод чрез забрана на прекъсванията:

При този подход в началото на критичната секция се забраняват всички прекъсвания, за да се осигури, че само текущият процес има достъп до ресурсите. В края на критичната секция прекъсванията се разрешават отново. Този метод беше типичен за първите еднoproцесорни операционни системи.

Spinlock (за многоядрени системи):

В съвременните паралелно работещи ядрови системи методът с забрана на прекъсванията не е приложим. Вместо това се използва допълнителен бит – наречен lock – който показва дали данните в критичната секция са свободни или се използват. При този подход процесът, който иска достъп до критичната секция, извиква процедурата spin_lock, която чрез операцията

test-and-set проверява и "заклучва" ресурса. След приключване на критичната секция ресурсът се освобождава чрез spin_unlock.

Пример за псевдоког:

spin_lock:

```
disable_interrupts()    // Забраняване на прекъсванията
R = test_and_set(lock)
if R == 0 then goto critical_section
enable_interrupts()     // Разрешаване на прекъсванията
goto spin_lock
```

critical_section:

```
// Изпълнение на критичната секция
```

spin_unlock:

```
lock = 0
enable_interrupts()    // Разрешаване на прекъсванията
```

Важно е този метод да не бъде рекурсивен и да се избягват прекъсвания по време на изпълнението на критичната секция, особено при драйвери, където се задават конкретни правила за управление на прекъсванията. Това е единствения механизъм от ниско ниво и е сравнително бърз. Недостатък е, че докато единия процес работи по критичната секция, другите процеси циклят, по-разумно е да бъдат спрени временно.

/etc – съдържа настройки на конкретната оперционна система. При инсталация там се записват описани конфигурациите на конкретната инсталация. Там е записана информацията като кои са потребителите, кои услуги са стратирани, как са настроени сърварите, които ще работят в тази машина

/home -съдържа файловете на потребители, за всеки потребител се прави поддиректория

/usr – където са програните, които потребителите ползват, документация, библиотеки

/var – данни, които са общи са потребителите и се обработват от програми, които обработват данните на много потребители, като бази данни. Там системата оставя съобщения за грешка, log файловете

/boot – поддиректория на /root, информация за драйверите

Тема 6

- Опишете понятията приспиване и събуждане на процес (block/wakeup).
- Семафор - дефиниция и реализация.
- Опишете разликата между слаб и силен семафор.
- Опишете накратко различните видове специални файлове в Linux:
- Външни устройства, именувани в /dev, псевдофайлове в /proc
- линкове - твърди и символни, команда ln
- сокеи

Тема 7

- ✓ *Взаимно изключване - допускане само на един процес до общ ресурс.*
- *Опишете решение със семафори.*
- *Качества и свойства на конкретните файловите системи, реализирани върху block devices.*
- *Ефективна реализация, отлагане на записа, алгоритъм на асансьора.*

Взаимното изключване означава, че само един процес може да използва критичната секция от началото до края, докато останалите процеси трябва да изчакат. Това се постига чрез различни алгоритми – например алгоритъма на Петърсен – както и чрез хардуерни решения, внедрени в съвременните компютърни архитектури.

Общо взето, за еднопроцесорните системи се използват два метода:

Метод чрез забрана на прекъсванията:

При този подход в началото на критичната секция се забраняват всички прекъсвания, за да се осигури, че само текущият процес има достъп до ресурсите. В края на критичната секция прекъсванията се разрешават отново. Този метод беше типичен за първите еднопроцесорни операционни системи.

Spinlock (за многоядрени системи):

В съвременните паралелно работещи ядрови системи методът с забрана на прекъсванията не е приложим. Вместо това се използва допълнителен бит – наречен lock – който показва дали данните в критичната секция са свободни или се използват. При този подход процесът, който иска достъп до критичната секция, извиква процедурата spin_lock, която чрез операцията test-and-set проверява и "заключва" ресурса. След приключване на критичната секция ресурсът се освобождава чрез spin_unlock.

Пример за псевдокод:

spin_lock:

```

disable_interrupts()    // Забраняване на прекъсванията
R = test_and_set(lock)
if R == 0 then goto critical_section
enable_interrupts()     // Разрешаване на прекъсванията
goto spin_lock

```

critical_section:

```

// Изпълнение на критичната секция

```

spin_unlock:

```

lock = 0
enable_interrupts()     // Разрешаване на прекъсванията

```

Важно е този метод да не бъде рекурсивен и да се избягват прекъсвания по време на изпълнението на критичната секция, особено при драйвери, където се задават конкретни правила за управление на прекъсванията. Това е единствения механизъм от ниско ниво и е сравнително бърз. Недостатък е, че докато единия процес работи по критичната секция, другите процеси циклят, по-разумно е да бъдат спрени временно.

Тема 8

- *Комуникационна тръба (pipe), която съхранява един пакет информация - реализация чрез редуване на изпращача/получателя.*
- *pipe с буфер - тръба, съхраняваща n пакета информация. Използване*
- *на семафорите като броячи на свободни ресурси.*
- *Права и роли в UNIX, команда chmod*
- *Роли - u/g/o user/group/others*
- *Права - r/w/x read/write/execute*

Тема 9

- *Взаимно блокиране (deadlock)*
- *Гладуване (livelock, resource starvation)*
- *Пример: задача за философите и макароните*
- *Единна йерархична файлова система в UNIX.*
- *Файлове и директории, команди - cd, mkdir, rmdir, cp, mv, rm*

Тема 10

- *Процеси в многозадачната система.*
- *Превключване, управлявано от синхронизация.*
- *Превключване в система с времеделене - timer interrupt.*
- *Опишете функционалността на следните команди в Linux:*
- *ls, who, find, ps, top*

Тема 11

- *Възможни състояния на процес. Механизми и структури за приспиване/събуждане.*
- *Диаграма на състоянията и преходите между тях.*
- *Опишете функционалността на следните команди в Linux:*
- *vi, tar, gcc*

Тема 12

- *Процес и неговата локална памет - методи за изолация и защита.*
- *Йерархия на паметите - кеш, RAM, swap.*
- *Виртуална памет на процеса - функционално разделяне (програма, данни, стек, heap, споделени библиотеки).*
- *Опишете функционалността на следните команди в shell:*
- *echo, read, test, if, for, while*

Тема 13

- *Таблицы за съответствието виртуална/реална памет.*
- *Ефективна обработка на адресацията - MMU, TLB.*

- файлови дескриптори, номера на стандартните fd, пренасочване
- филтри - cat, grep, cut, sort, wc, tr

Тема 14

- Избройте видове събития, причиняващи повреда на данните във файловите системи.
- Опишете накратко стандарта RAID5. Какво е журнална файлова система?
- СВързване и допускане до UNIX система - login.
- Конзола - стандартен Вход, стандартен изход, стандартна грешка.
- Команден интерпретатор - shell. Изпълнение на команди, параметри на команди

Тема 15

- Опишете разликата между синхронни и асинхронни Входно-изходни операции.
- Дайте примери за програми, при които се налага използването на асинхронен Вход-изход.
- Опишете с по едно-две изречения работата на следните системни извиквания в стандарта POSIX:
- socket(), bind(), connect(), listen(), accept()

Тема 16

- Опишете понятието “пространство на имената” (VFS).
- Структура, обекти и техните атрибути във VFS за ОС Linux.
- Основни функции, които обслужва пространството на имената.
- Една от класическите задачи за синхронизация се нарича “Задача за читателите и писателите” (readers-writers problem).
- Опишете условието на задачата и решение, използващо семафори.

Тема 17

- Опишете какви атрибути имат файловете в съвременна файлова система, реализирана върху блочно устройство (block device).
- Опишете накратко целта и реализацията на следните инструменти:
- (а) разместване във Времето на дисковите операции, алгоритъм на асансьора.
- (б) поддържане на буфери (кеширане) на файловата система.
- Опишете как се изгражда комуникационен канал (connection) между процес-сървер и процес-клиент със следните системни извиквания в стандарта POSIX:
- socket(), bind(), connect(), listen(), accept()

Тема 18

- Опишете накратко основните комуникационни канали в ОС Linux.
- Кои канали използват пространството на имената и кои не го правят?
- Опишете какви изисквания удовлетворява съвременна файлова система,

- *реализирана Върху блочно устройство (block device).*
- *Опишете предназначението на Журнала на файловата система.*