

# Структура на операционните системи

Любомир Чорбаджиев<sup>1</sup>  
lchorbadjiev@elsys-bg.org

<sup>1</sup>Технологическо училище “Електронни системи”  
Технически университет, София

20 октомври 2019 г.



# Съдържание

## 1 Функции на операционната система

## 2 Кратка история

- Пакетна обработка
- Многозадачност
- Времеделене

## 3 Основни компоненти на операционната система

- Управление на процесите
- Управление на паметта
- Входно/изходна подсистема
- Управление на външните запомнящи устройства
- Файлова система
- Механизми за защита и сигурност на информацията
- Команден интерпретатор

## 4 Архитектура на операционните системи

- Монолитни операционни системи
- Слоеста архитектура (многослойна архитектура)
- Микроядра

# Дефиниция за операционна система

- Няма общоприета дефиниция за операционна система.
- Операционната система може да се разглежда като:
  - програма, която управлява и разпределя ресурсите на компютърната система;
  - слой, който предоставя абстрактен интерфейс към хардуерните компоненти на компютъра.

# Основни функции на операционната система

- Предоставят начини за взаимодействие на потребителя с операционната система.
- Изпълнение на програми – възможността на операционната система да зарежда в паметта програми и да ги изпълнява.
- Входно/изходни операции – тъй като потребителските програми не могат директно да изпълняват входно/изходни операции, операционната система трябва да предостави средства за вход/изход.
- Манипулиране на входно/изходната система – предоставя възможност на потребителя да чете, пише, създава и изтрива файлове.
- Комуникации – размяна на информация между процеси, които се изпълнява върху един и същ компютър или на различни компютри свързани в компютърна мрежа.
- Обработка на грешки.

# Еволюция на операционните системи

- Ранните компютри не са имали операционни системи.
- Първоначално се появяват системи за последователна обработка на заданията.
- Операционни системи поддържащи многозадачност и времемелене.
- Персонални компютри.
- Разпределени системи.

# 1950–1960: Пакетна обработка

- Заданията се групират в пакети за последователно изпълнение.
- Системна програма *монитор*, която контролира изпълнението на заданията.
- Част от монитора винаги се намира в паметта – резидентен монитор.
- При стартиране на системата се зарежда монитора и управлението се предава на него.
- Работа на монитора е да зареди следващото задание и да предаде управлението на него.
- Когато дадена програма завърши своето изпълнение управлението се предава обратно на монитора.

# 1950–1960: Пакетна обработка

- Разработват се специализирани езици за управление на заданията (Job Control Language – JCL):
  - специализиран език за програмиране;
  - дава инструкции на монитора какъв компилатор да използва, от къде да вземе данните за програмата и т.н.
- Заедно с развитието на пакетната обработка се развива и хардуера, необходим за поддръжка на пакетна обработка на заданията:
  - защита на паметта – не позволява на работещите програми да променят областта от паметта, в която е разположен монитора;
  - таймери – предпазват системата от задания, които никога не свършват.

# 1960–1970: Многозадачност

- В ранните операционни системи е типична еднозадачната обработка:
  - в оперативната памет има само едно задание;
  - когато програмата извършва входно/изходна операция, процесорът чака тази операция да завърши.



**Фигура:** Еднозадачна обработка



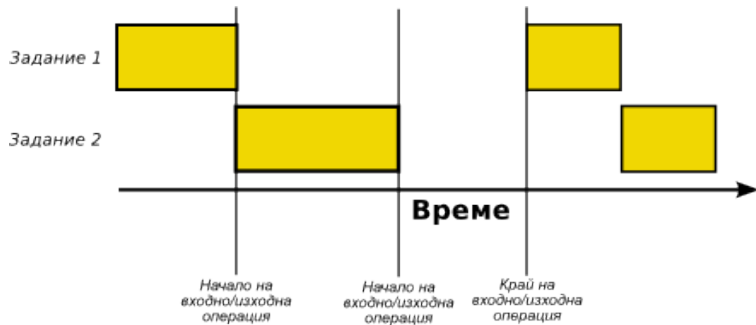
# 1960–1970: Многозадачност

- Процесорът е значително по-бърз от периферните устройства. Когато задачата е свързана с голям обем входно/изходни операции процесорът се използва много неефективно.
- С цел по-ефективно използване на процесора се преминава към многозадачна обработка:
  - в оперативната памет едновременно се намират няколко задания;
  - когато някое от заданията чака за изпълнение на входно/изходна операция, процесорът може да премине към обработване на друго задание.



**Фигура:**  
Многозадачност

# 1960–1970: Многозадачност



**Фигура:** Многозадачна обработка

# 1960–1970: Времеделене

- Организацията на работа при пакетната обработка е такава, че времето между подаването на заданието и получаването на резултатите е няколко часа.
- Желанието за по-кратко време на реакция води до възникването на идеята за времеделене - многозадачна работа, при която всеки потребител има терминал.
- Времето на процесора се разделя между всички потребители.

# 1960–1970: Времеделене

	Пакетна обработка	Времеделене
Основна цел	Оптимизация на използването на процесора	Минимизиране на времето за реакция
Въвеждане на команди	Командите на езика за управление на заданията се включват в заданияето	Командите се въвеждат от терминал

**Таблица:** Сравнение между пакетната обработка и времеделенето

# Управление на процесите

- Процесът е програма, която се изпълнява.
- Единица работа в рамките на операционната система.
- Програмата е пасивен обект, процесът е активен.
- Може да има няколко процеса, които изпълняват една и съща програма.

# Управление на процесите

- Операционната система изпълнява следните дейности, свързани с управление на процесите:
  - Създаване (create) и унищожаване (delete) на процеси.
  - Спиране (suspend) и възстановяване (resume) на процесите.
  - Предоставяне на механизми за синхронизация и комуникация между процесите.

# Управление на процесите

- Процесите се нуждаят от определени ресурси за да изпълнят задачата си: процесорно време, памет, файлове и входно/изходни устройства.
- Унищожаването на процеса изчиства и възстановява на всички ресурси, които той е използвал.
- Еднонишковите процеси имат само един програмен брояч, който определя положението на следващата инструкция.
- Многонишковите процеси имат по един програмен брояч за всяка нишка.
- Типично в една система има много процеси които се изпълняват конкурентно върху един или няколко централни процесора.

# Управление на паметта

- Паметта е голям масив от думи или байтове, всеки от които има собствен адрес.
- Паметта е склад за бързо и лесно достъпни данни, който се поделя между процесора и входно/изходните устройства.
- Оперативната памет е енергозависимо запомнящо устройство.
- Данните, които се съхраняват в оперативната памет се губят при повреда на системата.



# Управление на паметта

- Всички инструкции към процесора трябва да са в оперативната памет.
- Всички данни трябва да са в паметта, преди да се обработят от централния процесор.
- Всички резултати от работата на процесора са запазват в оперативната памет.
- Частта от операционната система, която управлява паметта, решава какво и кога да се съхранява в оперативната памет, като оптимизира използването на процесора.

# Управление на паметта

- Действията по управление на паметта са:
  - Да следи кои части от паметта от кой процес се използват.
  - Да решава кои процеси (или части от тях) и данни трябва да са в оперативната памет.
  - Да заделя и да освобождава части от паметта когато е нужно.

# Входно/изходна подсистема

- Входно изходната система се състои от:
  - Буферираща и кешираща система.
  - Общи интерфейси за драйвери на устройства.
  - Драйвери за конкретните хардуерни устройства на компютърната система.
- Една от задачите на операционната система е да скрие странностите на хардуерните устройства от потребителя.

# Входно/изходна подсистема

- Основна задача на входно/изходната система е управлението на паметта, използвана от устройствата:
  - Буфериране – памет, която се използва за временно запазване на данните по време на прехвърлянето им между устройството и оперативната памет.
  - Кеширане – запомняне на части от данните върху по-бързо запомнящо устройство, с цел подобряване на производителността.
  - Спулинг.

# Управление на външните запомнящи устройства

- Тъй като оперативната памет е енергозависима и твърде малка за да съхранява всички данни и програми, компютърната система притежава външни запомнящи устройства.
- Практически всички компютърни системи използват твърди дискове за да съхраняват данни и програми.
- Обикновено върху тях се запомнят данните, които не се събират в оперативната памет или трябва да се съхраняват голям период от време (между две пускания на системата).

# Управление на външните запомнящи устройства

- Операционната система отговаря за следните дейности по управление на дисковете:
  - Управление на свободното пространство върху дисковете.
  - Заделяне на пространство.
  - Планиране на дисковете.

# Управление на външните запомнящи устройства

- Правилното управление на външните запомнящи устройства е от централно значение за ефективната работа на операционната система.
- Скоростта на цялата компютърна система се крепи на дисковата подсистема и нейните алгоритми.
- Има външни запомнящи устройства, работата с които не е задължително да бъде бърза:
  - устройства за архивиране – оптични устройства, магнитни ленти и др.
  - различни типове – WORM (write-once,read-many-times) и RW (read-write).

# Управление на външните запомнящи устройства

- Операционната система предоставя единен, логически интерфейс към всички външни запомнящи устройства.
- Абстрактен слой, който скрива физическите хардуерни особености на запомнящото устройство и предоставя логическа единица за съхранение на данни – *файл*.
- Различните видове външни запаметяващи устройства притежават силно различаващи се качества по отношение на скоростта, капацитета, скоростта на предаване на данни, метода на достъп и др.



# Файлова система

- Файлът представлява съвкупност от данни, които имат връзка помежду си.
- Типично като файлове се съхраняват програми (в изпълним вид и във вид изходен код) и данни.
- Файловете типично са групирани в директории.
- Правата за достъп до файловете и директориите определят кой какво може да прави с файловете.

# Файлова система

- Операционната система отговаря за следните дейности, свързани с управлението на файлове:
  - Създаване и изтриване на файлове.
  - Създаване и изтриване на директории.
  - Поддръжка на примитивните операции с файлове и директории – писане, четене, преименуване, местене и т.н.
  - Устойчиво съхраняване на файловете и директориите върху външните запомнящи устройства.
  - Архивиране на файлове и директории.

# Защита и сигурност

- Защита се нарича всеки механизъм на операционната система, който контролира достъпа на процесите или потребителите до ресурси.
- Механизмите за защита:
  - трябва да могат да различат оторизиран и неоторизиран достъп до даден ресурс;
  - да позволяват дефиниране на правила за достъп до различните ресурси;
  - да предоставят средства за прилагане на дефинираните правила.
- Сигурността се дефинира като защита на системата от вътрешни и външни опити да се нарушат или заобиколят правилата за защита.

# Команден интерпретатор

- На операционната система могат да бъдат дадени голямо количество команди, отнасящи се до различни функции на операционната система, като например:
  - създаване и управление на процеси;
  - управление на входно/изходни операции;
  - управление на вторичните запомнящи устройства;
  - достъп до файловата система;
  - управление на механизмите за защита;
  - мрежови функции и т.н.
- Програмата, която чете и интерпретира командите на потребителя се нарича команден интерпретатор или шел (shell).
- Основната функция на командния интерпретатор (шел) е да прочете и изпълни командите на потребителя.

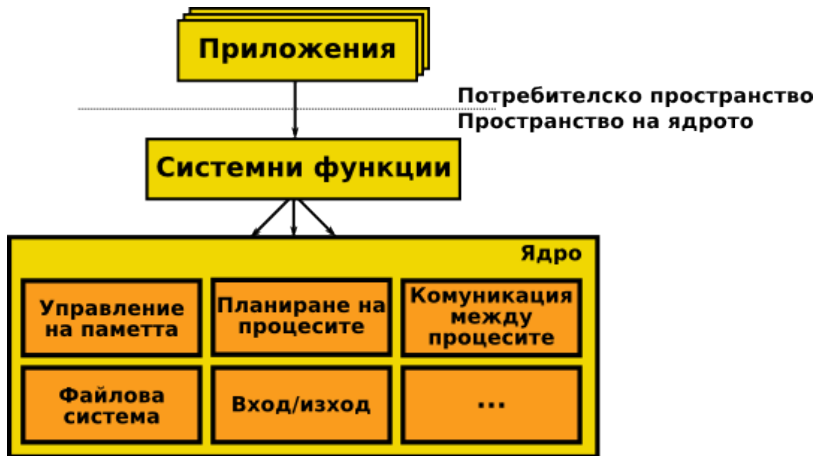
# Архитектура на операционните системи

- Съвременните операционни системи са сложни системи.
- В състояние са да предоставят голям набор от услуги на потребителите.
- Поддържат голямо количество изключително разнообразен хардуер.
- Архитектурата на операционната система дава възможност за справяне със сложността.
  - Разделяне на операционната система на компоненти.
  - Определя функциите и възможностите на всяка компонента.

# Монолитни ядра

- Типична организация на ранните операционни системи.
- Всяка компонента на операционната система се съдържа в ядрото.
- Всяка компонента на операционната система може директно да комуникира с всяка друга компонента.
- Висока производителност – няма забавяне за комуникация между компонентите на операционната система.
- Много трудно се проследяват бъгове. Грешки в една компонента могат да доведат до нежелани ефекти в друга компонента.

# Монолитни ядра



Фигура: Монолитна операционна система

# Ранен UNIX

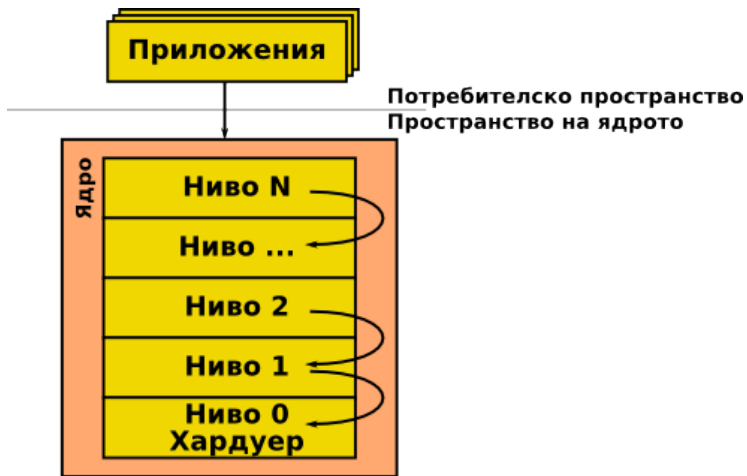
- Ранният UNIX притежава ограничена поддръжка за хардуер.
- Структурата е проста: монолитно ядро и системни програми.
- Ядрото:
  - Съдържа всичко, което е между интерфейса на системните функции и хардуера.
  - Предоставя файлова система, планиране на процесора, управление на паметта и т.н.



# Многослойна архитектура

- Опит за подобряване на монолитната архитектура.
- Групира компоненти с подобна функционалност в слоеве.
- Операционната система е разделена на определен брой от слоеве, като всеки слой се изгражда върху основата на по-долните слоеве.
- Най-долното (най-ниското) ниво (ниво 0) е хардуера.
- Най-горното ниво (ниво N) се състои от потребителския интерфейс и потребителските програми.

# Многослойни ядра



**Фигура:** Многослойна операционна система

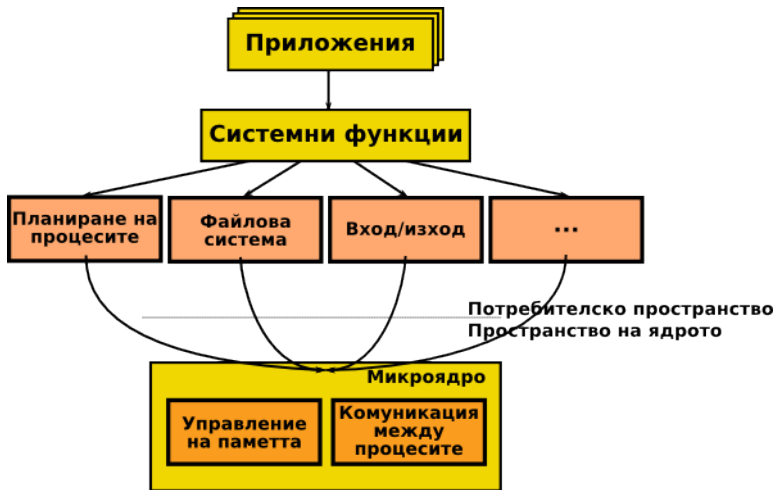
# Многослойна архитектура

- Всеки слой е организиран така, че да използва функции и услуги само от по-долните нива.
- Даден слой може да комуникира само със съседните слоеве.
- Заявка за изпълнение на някаква операция преминава последователно през слоевете, докато достигне до слоя, който може да я изпълни.
- Защита между слоевете.
- Намалява бързодействието на ядрото заради нуждата от организиране на комуникация между слоевете.

# Микроядра

- Опит ядрото да се направи малко.
- Ядрото предоставя само малък брой услуги – управление на паметта и комуникация между процесите.
- Много висока степен на модулност.
- Операционната система става лесно преносима и мащабируема.
- Нараства количеството на комуникацията между процесите, което намалява производителността на ядрото.
- Максимално количество от услуги, които типично са услуги предоставяни от ядрото, се преместват в потребителското пространство.
- Операционната система става по-надеждна тъй като по-малко код работи в незащитен режим.

# Микроядро

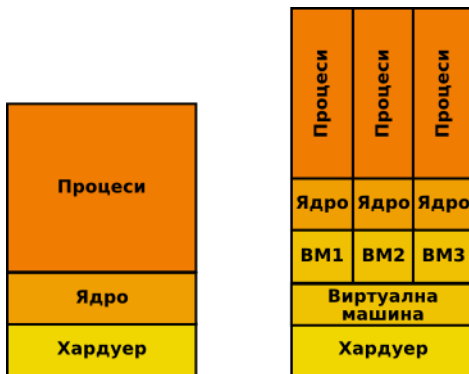


Фигура: Микроядро

# Виртуални машини

- Операционната система създава илюзията за много процеси, всеки от които се изпълнява върху свой виртуален процесор, своя виртуална памет и т.н.
- Ресурсите на компютърната система се използват за създаване на виртуални машини.
- Виртуалната машина предоставя интерфейс, който е идентичен с хардуерната платформа.
- Виртуалната машина предоставя пълна защита и контрол на ресурсите на системата.
- Всяка виртуална машина е изолирана от всички останали виртуални машини.
- Виртуалните машини са много удобни при разработване на операционни системи.

# Виртуални машини



**Фигура:** Виртуална машина

# Работа на операционната система

- Работата на операционната система се управлява от прекъсванията.
- Софтуерните грешки (делене на нула, неправилна адресация и т.н.) и заявките към операционната система генерират софтуерни прекъсвания (trap).
- Работата в два режима позволява на операционната система да се пази от другите компоненти на системата.
- Режимът на работа на процесора позволява да се прави разлика между потребителско приложение и ядрото на операционната система.
- Извикването на системна функция води до промяна на режима на процесора от потребителски към режим на ядрото.
- Когато изпълнението на системната функция приключи, режима на процесора се превключва на потребителски и изпълнението се връща на потребителската програма.



# Системни функции

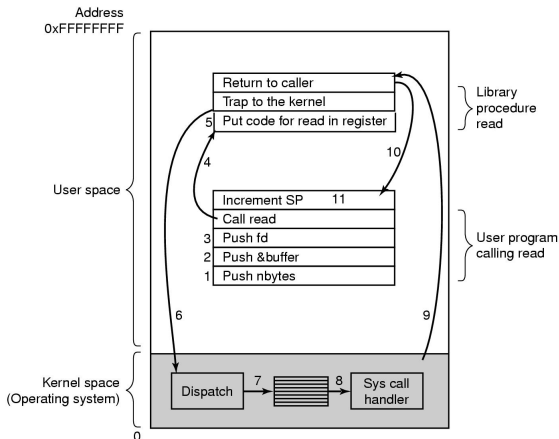


Фигура: Извикване на системна функция

# Системни функции

- Системните функции предоставят интерфейс между потребителската програма и операционната система.
- В общия случай системните функции са достъпни за програми на асемблер или други езици за системно програмиране като C и C++.
- Има различни начини за предаване на параметри на системна функция:
  - Параметрите се предават в регистрите на процесора.
  - Параметрите се запазват в област от паметта, чийто адрес се предава в някой от регистрите на процесора.
  - Параметрите се слагат в стека от програмата, и се вадят от стека от операционната система.

# Системни функции



**Фигура:** Извикване на системна функция (А. Таненбаум)

# Видове системни функции

- Функции за управление на процесите – `fork()`, `waitpid()`, `execvpe()`, `exit()`.
- Функции за работа с файлове – `open()`, `close()`, `read()`, `write()`, `lseek()`.
- Работа с файловата система – `stat()`, `mkdir()`, `rmdir()`, `link()`, `unlink()`, `mount()`, `umount()`, `chmod()`.
- Междупроцесна комуникация – `kill()`.