**ОПРЕАЦИОННИ СИТЕМИ - РАЗВИТИЕ И СЪВРЕМЕННОТО ИМ СЪСТОЯНИЕ**

Д-р инж.Мартин Пъшев Иванов,

главен асистент, НБУ –департамент „Информатика”,

e-mail: mivanov@nbu.bg

**Анотация**: В публикацията се описват основните концепции на операционните системи с общо предназначение, проследява се развитието и приложимостта им, очертава се структурата им и функционалността на основните компоненти: процеси, управление на паметта, управление на външните устройства, файлова система и пр. Описват се и особеностите на разпространените съвременни операционни системи – Windows, Linux и операционната система за мобилни устройства Android. Изложението носи образователен характер и е ориентирано към студенти от бакалавърските програми „Информатика” и „Информационни технологии”.

**Ключови думи**: *операционни системи, управление на процесите, управление на паметта, мобилни операционни системи, Windows, Android.*

1. **Въведение в проблема**

Съвременните компютри предлагат впечатляваща изчислителна мощност, гъвквост и функционалност, които са постигнати чрез продължителна техническа еволюция и революционни иновационни решения. Компютърната система (КС) е изключително сложно техническо изделие, съдържащо огромен брой взаимодействащи си компоненти. Той се състои от един или повече процесора, оперативна памет, свързващи шини, дискови подсистеми, разнообразна периферия и устройства за въвеждане и извеждане на информация. За да се осигури нормалната и пълноценна работа на изчислителното устройство и за да бъде то в състояние да предоставя очакваната от него функционалност, взаимодействието между съставящите го компоненти следва да се извършва в определен ред и да се подчинява на строго определени правила. Непосредственото управление на работата на такава сложна техническа структура е невъзможно, поради което се налага нейното функциониране да се управлява от специализирани софтуерни системи, определящи условията на работа на съставящите я технически компоненти, връзките и зависимостите между тях.

Понятието „операционна система” (ОС) не винаги е ясно определено като термин и в значителна степен зависи от равнището на развитие на конкретната компютърна система и обхвата на изпълняваните от нея задачи. Операционната система най-общо може да се определи като софтуерен инструмент за управление на ресурсите на КС. Тя управлява работата на централния процесор, на оперативната памет, на устройствата за вход/изход (В/И), както и на другите компоненти на компютъра. Една от основните функции на ОС е да създаде равнище на абстракция, което “скрива” от потребителя сложната структура и функциониране на апаратните средства на КС, да представи достъпа до тях и използването им в удобна форма на абстрактни логически концепции.

Операционната система е компютърна програма, която контролира работата на приложните програми и системните приложения и изпълнява ролята на интерфейс между приложенията и апаратната част на компютъра ((Tanenbaum,2009), (Silberschatz,2006), (Silberschatz,2012)). Нейното предназначение може да бъде представено в две основни направления:

* + - **Приложен аспект** – като интерфейс между потребителя и компютърната система, създаващ максимално удобство и полезност за потребителя на компютърната система – ОС прави използването на компютърната система достъпно, удобно и производително.
		- **Системен аспект** – като диспечер и мениджър на ресурсите на компютърната система, осигуряващ ефективното изпълнение на системните и приложните програми. Ефективност – ОС съдейства за ефективното използване на ресурсите на компютърната система.

Първият аспект на предназначението на ОС акцентира върху разбирането, че тя представлява програма, която функционира като необходим посредник между потребителя на компютърната система и нейната апаратна част. Нейното предназначение в този случай се разбира като предоставяне на достъпна и удобна среда за потребителя и средства за подобряване на производителността на неговата работа. Приложният аспект предполага, че ОС е представена пред потребителя като система от абстракции, скриващи детайлите и техническите особености на компютърната архитектура. Следва да се вземе под внимание, че абстракциите се отнасят преди всичко до категориите, описващи дейностите по изпълнение на потребителските задачи и зависят от типа на потребителите на КС. Самите потребители на компютърната система са няколко типа, имащи различна подготовка, мотивация и решаващи различни задачи:

* + - **Крайни потребители** – използващи ресурсите на ОС в рамките на приложенията, които ползват за практическите си задачи;
		- **Разработчици и програмисти** – нуждаят се от достъпен и удобен програмен интерфейс към функциите на ОС (библиотеки от системни функци, системни извиквания, достъп до драйвери), а също и от развойни средства и инструменти за тестване и настройване на разработваните приложения.
		- **Системни специалисти и администратори** – приоритетни за тях са удобни и многофункционални инструменти, които позволяват да се работи със структури и параметри на ОС и на основните системни приложения на различни равнища.

В случаите на използване на персонални или мобилни компютърни устройства, това разделяне на потребителите е твърде условно и се изразява по-скоро в характера на изпълняваните задачи от ОС към даден момент. Съществено място в осмислянето на приложния аспект на ОС е това на потребителския интерфейс, който може да бъде решен на ниво интерпретация на текстови команди или пък чрез усъвършенствани многофункционални интуитивни визуални средства.

Системният аспект на функциониране на ОС остава скрит за крайния потребител, но той има решаващо значение за ефективното и пълноценно използване на ресурсите на КС. Основните задачи на ОС на това равнище са:

* + - Организация на ефективното взаимодействие на всички компоненти, съставящи компютърната система в процеса на изпълнение на програмите, решаващи потребителските задачи (системен аспект). ОС организира и следи за разполагането на потребителските програми и необходимите за работата им данни в оперативната памет и върху вторичните носители на на компютърната система, осигурява достъпа до и комуникацията с необходимите устройства.
		- Осигуряване на на т.нар. типови ”системни задачи”, който е необходимо да бъдат изпълнени по заявка и в интерес на потребителя – достъп и работа с файловата система, създаване, изтриване и възстановяване на файлове, достъп до периферни устройства, специфични потребителски задачи и т.н. Обикновено тези дейности се осъществяват от специализирани програмни модули, наречени „помощни програми” или utility.

Ресурсите на КС са разнообразни, свързани в сложна техническа структура, като всеки от тях има своето функционално място и предназначение в изпълнението на потребителските задачи. За да бъдат ефективно и целесъобразно използвани, те трябва да бъдат активирани в определен ред с отчитане на необходимите логически и технологични връки на тяхното ваимодействие. Основните ресурси, обект на управляващите функции на операционната система са: оперативната памет, процесорното време, вторичната памет, разположена върху периферийни устройства, евентуално – мрежови комуникации. Самите управляващи функции на операционната система по отношение на ресурсите се отнасят до:

* + - Реализацията на ефективен достъп на ресурсите от страна на използващите ги програми;
		- Ефективното им споделяне в многозадачен режим и в условия на колективно използване – с необходимите средства за осигуряване на желаната производителност на КС, защита, сигурност и безопасност.
		- Обработка на критични ситуации и грешки и средства за възстановяване при настъпването им.

По своята същност операционната система представлява нормална компютърна програма, съставена и реализирана в инструкциите, които може да разпознава и изпълнява апаратната част на компютъра. По силата на координиращите и управляващите функции на операционната система някои от инструкциите ѝ се изпълняват в особен „привилегирован” режим, наречен още „режим на ядрото” (kernel mode), който е недостъпен за кода на обикновените потребителски приложения. Привилегировани инструкции например са: директен достъп до В/И, мрежови устройства и контролери, управление на таблиците на паметта и достъп до защитени нейни области, манипулиране на системни структури и битове на състоянията, обработка на прекъсанията и др. Потребителските програми работят в т.нар.”потребителски режим” (user mode). Те могат да ползват привилегировани инструкции при необходимост само индиректно, с изключителното посредничество и контрол на ОС, използвайки специализиран програмен интерфейс, наречен „системни извиквания” (system calls). Самата част от кода на ОС, която отговоря за базовите ѝ и критични управляващи функции се разполага резидентно (постоянно, през цялото време на работа на КС) в оперативната памет и се нарича „ядро” на ОС (kernel).

1. **История и развитие на операционните системи**

Краткият исторически обзор ще даде възможност да се представи динамиката на развитие на идеята на операционните среди и системи, възникването и развитието на основните им принципи и концепции, тенденциите в развитието, условията и особеностите при които са с оформили техните структурни и функционални характеристики и да се оцени сложният и дълъг път до достигането на съвременното им състояние. Основните фактори, които оказват влияние върху развитието на ОС са:

* + - Разширяване и усъвършенстване на съществуващия хардуер, както и създаването на нов такъв;
		- Предоставянето на нови фукционалности и услуги в съответствие с нарастащите потребности на потребителската среда.
		- Отстраняване на регистрирани проблеми и програмни грешки.

Хронологично развитието на операционните системи може да бъде систематизирано в няколко поколения, всяко от които отбелязва съвкупните характеристики и особености на съответния исторически период:

**Първо поколение** – то отбелязва най-ранния период около 40-те години на изминалия век, свързан с първите опити да се създаде автоматична изчислително устройство на базата на електронна техника. Важен исторически момент е създаването на първия електронен компютър ABC от проф.Джон Атанасов (John Vincent Atanasoff) и неговия сътрудник Клиффорд Бери (Clifford Edward Berry) – 1930 - 1936 г. Хардуерната част на компютъра е била реализирана на базата на около 300 електронни лампи и е била в състояние да изпълнява прости последователности от аритметични операции. Появилите се впоследствие други технически реализации на изчислителни системи принципно преповтарят и разширяват основната идея на програмираните изчисления. Техническите решения са примитивни, работата с тях е по-скоро в областта на научните изследвания и експерименти. Потребителите на тези „прото-компютри” използват чисто технически средства, за да въведат програмния „код” (представен чрез състоянията на двоични превлючватели, светлинни индикатори и командни пултове) и са далеч от мисълта да търсят апаратни или софтуерни решения за някакво цялостно управление на работата на изчислителната система. Работата с компютрите на този етап е трудоемка, изискваща много ръчни операции по подготовка на техническите устройства, слабо ефективна, ненадеждна. свързана с чести откази на апаратната част.

**Второ поколение** – исторически то се свързва с началото на 50-те години на миналия век, когато вниманието към използването на изчислителни системи се засилва и се появяват очаквания за приложението им при решаване на сериозни и тежки изчислителни задачи. Този период е свързан със стремежа да се подобри надеждността на изчислителните устройства, да се опростят средствата и техниките за въвеждане на данни и програмен код, да се облекчи и опрсти работата на специалистите, свързани с експлоатацията на тези устройства. Съществено е и изискването да се постигне максимална ефективност при използването на изчислителната система и свързаната с използването ѝ висока цена на показателя „машинно време”. През този период се появяват първите идеи за обединяване на съвкупност от семантично свързани дейности в единно „задание” (job), както и за съставяне на графици за изпълнение на тези задания от кокретната компютърна система. Изчислителните устройства са в състояние да изпълняват само една единица работа в отделен момент от времето, т.е. те предлагат еднопрограмна последователна (пакетна) обработка. Последователните задания се оформят в „поток”, работата на компютърната система се управлява от прости програми, наречени „монитори”, които са отговорни преди всичко за зареждането на кода на заданията и завършването на работата им. Други характерни компоненти на системното програмно осигуряване през този период са специализирани зареждащи програми (loaders), драйвери на устройствата (с пряко адресиране), модули за обработка на грешки и др. Това е и периодът, през който се появяват първите компилатори за езици от високо ниво (FORTRAN, COBOL, ALGOL и др.). При този режим на експлоатация крайният потребител не се нуждае от пряк контакт с изчислителната система, а я използва чрез „пакетна” (batch) организация на работа, администрирана от специализирания обслужващ персонал.

**Трето поколение** – той е свързан с развитието на идеята за колективно използване на големи и скъпоструващи изчислителни системи чрез постигането на максимална ефективност и приемливи икономически параметри. Периодът е характерен и с разширяването на изчислителната мощност на устройствата, преминаването към нова транзисторна елементна база и интегрални схеми с ниска степен на интеграция. Типичната технология на работа с компютрите от това поколение е предимно пакетната обработка, но тя е разширена, появява се езиково средство за управление на работата със системата – команден език (JCL – Job Control Language). Разширена е и апаратната част компютърната система и средствата за софтуерното ѝ управление – появяват се входно/изходните процесори (канали), механизмът на програмните прекъсвания, някои основни елементи на мултипрограмирането, наличието на резидентно ядро на управляващия софтуер и резидентни модули – т.е. първоначална, макар и значително опростена форма на идеята за операционна система. Други разширения на средствата за управление на работата на компютърната система са модулът за първоначално зареждане (IPL – Initial Program Loader), както и на възможността за спулинг (SPOOL – *Simultaneous Peripheral Operations On Line*) – паралелно изпълнение на операциите за вход/изход чрез специализирани канали, независимо от работата на централния процесор.

**Четвърто поколение** – характеризира се с окончателно реализиране на идеята за съвместно паралелно изпълнение на няколко задания (с фиксиран или променлив брой) – т.нар.”мултипрограмиране”. Този период се свързва със 70-те години на изминалия век и със значително радикално усъвършенстване на елементната база – появата на големи и свърхголеми интегрални схеми, с висока степен на интеграция, с намалена консумация на енергия и понижена цена. Компютрите стават масово разпространени, но все още ползвани в условията на колективна работа и предимно пакетна обработка. През този период системите за управление на компютърните устройства се оформят като пълноценни и сложни операционни комплекси, поддържащи собствен команден език, средства за планиране, диспечиране и управление на процесите, средства за управление и зашита на паметта, средства за поддържане и управление на разнообразна периферия и комуникации и др. Операционните системи от този период предоставят много потребителски функции, много функции за изпълнение на системни задачи и за достъп до системни ресурси, административни функции за контрол на достъпа, защита, сигурност и безопасност.

**Пето поколение** – в значителна степен този период се припокрива хронологично с периода на съществуване и експлоатация на компютрите от предишното поколение. Той е свързан е появата и развитието на микропроцесорната техника и с радикална промяна в достъпа на крайния потребител до компютърната система. Техническото и технологично равнище на електронната техника през този период значително понижават цената на компютърните системи при революционно нарастване на тяхната изчислителна мощност, памет, разнообразие в използваната и поддържаната периферия. В началото на периода се появяват и първите операционни системи, предназначени за новото поколение компютри като CP/M, MS-DOS, а така също и ранните версии на ОС Unix, а краят не периода (продължаващ и до днес) се ознаменува от развитието на мощни многофункционални операционни системи, съответстващи на върховите постижения на техническия прогрес и най-съвремените виждания за мястото на компютърните технологи във всички срани на разнообразната икономическа и социална дейност на човека. Основните акценти в съвремените операционни са два – максималнa достъпност, удобство и полезност за работата на потребителите с различна подготовка в областта на компютърните технологии и максимална ефективност при работата с разнообразни компютърни архитектури и конфигурации. Появяват се средствата за мрежова комуникация, работа с многопроцесорни и многоядрени ахитектури, технолгии от типа „клиент/сървър”, използването на споделени и Web-базирани ресурси, многослойни приложения, Web-услуги и др. Важно е да се отбележи, че поради високата динамика на развитие на електронните технологии и софтуерните средства, както и поради непрекъснато раширяващият се обхват на възможните приложения и решаваните задачи този период може да бъде обхванат в едно поколение единствено условно. Внимателният анализ на развитието на компютърната техника и на софтуерните решения през този период може да покаже наличието на специфична за него хронологична структура, значително по-сложна и развита от направеното тук общо разделение на пет основни генерации. Така например съществени и ясно различими подгрупи от компютърни устройства в групата на поколението, отбелязано като „пето” са тези на преносимите компютри и персоналните мобилни компютърни и комуникационни устройства.

В така направения исторически преглед на развитието на компютърните устройства мнозина отбелязват две свързани тенденции – ранният етап на използване на изчислителните устройства се отбелязва от някои изследователи като период на „скъпи компютри и евтин специализиран труд”. Съвременният исторически етап на приложение на компютърните средства и технологии обръща това отношение във формата „евтини компютри – скъп специализиран труд”.

1. **Основни функции на ОС**

Класическото определение на основните функции на операционните системи, осигуряващи реализацията на приложния и системния им аспекти включва ((Tanenbaum,2009), (Silberschatz,2006), (Silberschatz,2012)):

* + - Управление на процесите;
		- Управление на паметта;
		- Управление на външната (втричната) памет;
		- Управление на входа и изхода (В/И);
		- Създаване, достъп и поддържане на файловата система;
		- Поддържане на мрежи и комуникации;
		- Средства за защита и сигурност на процеси и данни;
		- Потребителски интерфейс.

Най-съществените от тях ще бъдат разгледани самостоятелно в изложението в рамките на необходимия за разбирането им обхват и детайли.

1. **Основни компоненти на ОС**

Структурата на ОС е съставена от модулите, които осъществяват основните ѝ функции. Те в най-общия случай са ((Tanenbaum,2009), (Silberschatz,2006), (Silberschatz,2012), (Николов,2012)):

* + - Ядро на операционната система (kernel) – базовият модул, чиито код работи в т.нар. „привилегирован” режим и е отговорен за най-съществените функции на ОС – управление на зареждането и изпълнението на програмите (процесите), управление на паметта, достъп до файловата система, обмен на файлове с периферните устройства и др.;
		- Средства на потребителския интерфейс („потребителска обвивка”) -команден процесор, интуитивен графичен интерфейс (Graphic User Interface) и всички други средства за комуникация на ОС с потребителя;
		- Драйвери на периферийните устройства – отговарящи за управлението на тези устройства и за осигуряване на достъпа до представяните от тях данни;
		- Допълнителни обслужващи програми (utilities) – изпълняващи типови и специфични полезни функции по използването и поддържането на ОС. Тук могат да се добавят и разнообразните динамични библиотеки (Dinamic Link Libraries - DLL), използвани от различните приложения и трансформиращи обръщенията към системните функции в обръщения към модулите на ядрото.

Според типа, предназначението и конкретната реализация на ОС тези модули могат да имат свое детайлно вътрешно развитие, да бъдат допълвани и модифицирани. Във всички случаи те трябва ефективно да осигуряват изпълнението на основните функци на ОС. Примерна концептуална структура на ОС е изобразена на фиг. 1 .



Фиг. 1. Концептуална структура на ОС.

1. **Класификация на операционните системи**

Класификационните признаци на ОС са разнообразни. Една изходна класификация на операционните системи може да бъде направена, разбира се, в съответствие с отбелязаните по-горе пет поколения в тяхното историческо развитие, но значително по-популярни са други два класификационни критерия, считани за основни:

* + - броят на едновременно изпълняваните задачи в компютърната система – еднозадачни и многозадачни;
		- броят на потребителите, които могат едновременно да ползват компютърната система – еднопотребителски и многопотребителски.

Броят на едновременно изпълняваните задачи в компютърната система е свързан преди всичко със степента на техническото развитие на нейната апаратна част, но от съществено значение е наличието и на съответни софтуерни компоненти на ОС, които могат успешно и ефективно да реализират тази възможност. В съвременните операционни системи това се постига с механизмите за планиране и управление на работата на централния процесор (респ.процесори).

**5.1 Сървърни операционни системи** – предназначени за обслужване на мощни високопроизводителни сървърни компютърни системи, предназначени за колективно ползване и достъп до общи ресурси, с ограничен потребителски интерфейс и с приоритетно развити функции по ефективно управление на комуникацията с клиентите, поддържане на необходимите комуникационни протоколи, регистрация и отчет на потребителите, управление на достъпа, надеждна защита на данни, сигурност и безопасност.

**5.2 Операционни системи за „реално време”** (Real Time Operating Systems - RTOS) – това са ОС, които осигуряват работата на специализирани изчислителни устройства, следящи или управляващи динамично протичащи процеси. Основната особеност на тези ОС е способността да осигурят бърза реакция на системата за определен период от време с критична продължителност (секунди, милисекунди и т.н.). Такива устройства се използват в автоматични навигациони системи, сложни и динамични технологични процеси, системи, управляващи транспортния трафик, специализирани военни, авиационни и космически приложения. Като приоритетен критерий за разработване на ОС от този тип служи критичното време за отговор, което може да е за сметка на точността на обработваните резултати.

**5.3 „Вградени” операционни системи (Embedded Operating Systems)** – това са специализирани системи с ограничен или липсващ потребителски интерфейс, предназначени да работят напълно автоматично и независимо от другите компоненти на технческата система. Обикновено тези системи решават една или няколко основни задачи, свързани с автоматичния режим на устройството, съобразени са с неговите архитектурни особености и хардуерни ограничения, и физически се разполагат върху енергонезависимо устройство с постоянна памет (от типа Read Only).

**5.4 Операционни системи за smart-карти** – малки ОС, предназначени за различни категории интелигентни бизнес карти, които имат малка памет и прост процесор. В повечето случаи функциите, които могат да изпълняват, са ограничени от една до няколко – свързани с изпълнението на плащания, проверки на банкови сметки и др. Съществуват и многофункционални ОС от този тип с разширени функции, a също и такива, които позволяват някаква форма на мултипрограмен режим. В някои случаи към програмното осигуряване на системи от този тип влизат и софтуерни компоненти, написани предимно на езика Java.

**5.5 Операционни системи за мобилни устройства** – това е една категория ОС, които получават изключително интензивно развитие в последно време както по обхват на апаратната част, чието функциониране осигуряват (това са разнообразните смартфони, таблети и др.), така и по своята структура и състав, области на приложение. Типичните особености на мобилните устройства, които оказват влияние върху архитектурата и функционалностите на този клас операционни ситеми с добре известни – това са разнообразния хардуер, разнообразните устройства за въвеждане, извеждане, за съхраняване на информация, средствата за комуникация, наличието на сензори, ограниченото захранване, известни, макар и относителни, ограничения в ресурсите, приложение и използване в различни областти, в различни режими от потребители с различна мотивация и подготовка. Популярността на тези операционни системи е много висока и постоянно нарастваща, броят на реализираните видове и варианти на тези ОС е много голям и се увеличава непрекъснато. Устройства с ОС от този тип имат много висока степен на функционална и структурна организация. Те станаха неразделна част от нашето ежедневие и се превърнаха в неотменим елемент на социалната среда. Следва да се отбележи, че това е едно от най-бурно развиващите се направления в операционните системи, което вероятно в недалечно бъдеще ще предостави на потребителя много иновативни решения и впечатляващи изненади.

1. **Архитектури на ОС**

Основната класификация на ОС според архитектурните концепции на разработването им включва следните групи:

* + - ОС с монолитно ядро - MS-DOS, Linux, FreeBSD;
		- ОС с многослойна архитектура (Layered approach architecture) -;
		- Архитектура на ОС с микроядро (microkernel architecture) - Mach, Symbian, MINIX 3.

**6.1 ОС с монолитно ядро** – това е една много стара и твърде разпространена концепция за реализацията на ОС. При нея ОС е представена като една голяма програма, която съдържа цялата функционалност в един голям блок от код. ОС се изпълнява като отделен процес в отделно адресно пространство. Монолитната ОС няма своя вътрешна структура, функционалната ѝ организация е изпълнена под формата на множество вътрешни процедури. Всяка от процедурите се извиква чрез дефиниран за целта интерфейс и има достъп до всички други процедури, както и до всички вътрешни структури от данни на операционната система. Непосредственото извикване на вътрешните процедури на монолтното ядро води до значителна производителност на ОС, но същевременно системата става твърде обемиста, изразходва значително памет като ресурс и е уязвима на вътрешни откази и грешки. Характерна е за ранни операционни системи със сравнително проста организация и минимални изисквания по отношение на сигурността и ефективното използване на ресурси (първи варианти на DOS и Unix).

**6.2 ОС с многослойна структура** – ОС е организирана като съвкупност от няколко послеователно подередени слоя. Най-ниският слой е този на хардуера (номер 0). Най-високият слой е този на потребителския интерфейс. Функционалностите са разпределени между слоевете на програмния стек. Всеки от по-високите слоеве ползва функциите и услугите на по-ниските, които осигуряват условията за неговата работа. Основното предимство на този тип архитектура е нейната модулност, опрстено програмиране, опростено тестване, лесно добавяне на нови функционалности и модули, лесно модифициране и настройка към промени в хардуера. Типична оганизация на ОС с многослойна структура е оказана на фиг. 2 .



 Фиг. 2. Организация на ОС с многослойна архитектура.

**6.3 ОС с архитектура на микроядро (клиент-сървър структура)**

Това е концепция за поместване на по-голяма част от кода и функционалностите в по-горните равнища на софтуерния архитектурен стек на ОС, оставяйки минимално по размер ядро (микроядро – *microkernel, μK*), Микроядрото може да бъде разглеждано като компактно ядро, което изпълнява само най-основните функции, универсални за всички компютри. То е съдадено, за да може да бъде интегрирано в различни ОС и работи със специфични ОС-сървъри, които осигуряват функциите от по-високо ниво. По-голяма част от функциите на ОС се реализират като потребителски процеси. Само кодът на ядрото може да съдържа “привилегировани операции” (изпълнявани в режим „kernel mode”). Обменът на данните между модулите става на базата на съобщения (*messages*) – специфични структури от потребителски и системни данни. Използването на микроядро е предпоставка за по-надеждна и по-сигурна ОС, с модулна организация и с по-лесна поддръжка.

**6.4 Виртуални машини**

Концепцията на виртуалните машини (ВМ) представлява специфично приложение и развитие на многослойната архитектура. Виртуалната машина представлява самосъдържаща се операционна среда, която има поведение на отделен компютър. Виртуалната машина емулира работата на конкретна компютърна система (реална или хипотетична), като се базира на нейната архитектура и функции. Виртуалната машина се изпълнява като самостоятелен процес на базовата компютърна система. Често реализирането ѝ изисква наличието на специализиран софтуер и хардуер. Използването на виртуални машини дава две основни предимства:

* + - **Системна независимост** – виртуалната машина работи по един и същ начин върху всяка реална система, независимо от хардуерните ѝ особености и софтуерната среда;
		- **Сигурност** – виртуалната машина няма непосредствен контакт с операционната система, поради което възможностите за увреждане на файлове и приложения са минимални. В този случай, обаче, трябва да се има предвид, че програмите, изпълнявани във виртуалната машина са отделени от операционата система, те не могат да ползват специфичните възможности на последната.

Работата на виртуалните машини в една операционна среда се управлява от специален модул, наречен „монитор на виртуалните машини” (Virtual Machine Monitor - VMM). Виртуалните машини имат своя класификация според степента, с която реализират функционалностите на целевата компютърна система. Използването на ВМ прави възможна работата на няколко различни ОС едновременно на една реална машина. Типичен популярен пример за изплзване на виртуална машина е средата за изпълнение на Java-програми (JRE - Java Runtime Environment). В този случай се осигурява преносимост на софтуера, написан на Java и напълно идентичното му (и безопасно!) изпълнение, независимо от особеностите на конкретната операционна система и хардуерна част.

1. **Основни концепции в структурата и функционирането на ОС**

Следва да се вземе под внимание, че операционната система е сложен софтуерен продукт, реализиран в разнообразни концепции и архитектури (Николов,2012). Тук ще бъдат отбелязани само основните концепции в структурата и функционирането на една ОС: понятието „процес”, понятието „памет” (с особеностите на своята йерархична организация), управлението не В/И устройства и др.

**7.1 Процеси**

Ключово понятие, което помага да се разбере структурата и функционирането на ОС, във всички операционни системи е **процесът (**Tanenbaum,2009**).** Този термин е въведен първоначално през 60-те години на миналия век при разработването на операционна система Multics и от тогава се използва повсеместно. Под термина „процес” обикновено се разбира програма по време на изпълнението ѝ в компютърната система, но следва да се вземе под внимание, че такова определение е твърде общо и непълно. По-коректно това понятие може да бъде определено като единица дейност, представена за изпълнение в компютърната система, обикновено решаваща някаква завършена потребителска или системна задача и изискваща изпълнението на поредица от действия, както и достъп до определена съвкупност от ресурси. Процесът се възприема и управлява от ОС като единно цяло. Той разполага със собствено адресно пространство – списък от адреси в паметта, които процесът може да използва за своите цели - обикновено недостъпно за другите процеси. Следва да се обърне внимание, че понятията „програма” и „процес” не бива да се смесват. В една многопотребителска среда една и съща програма може да се изпълнява от гледна точка на целите на различни потребители, но всяко конкретно изпълнение на програмата протича в рамките на отделен, напълно самостоятелен процес. Процесът може да създава свои наследници – дъщерни процеси и подпроцеси.

Типични задачи на ОС, свързани с управлението на процесите са следните:

* + - Създаване и прекратяване на процесите;
		- Превключване на процесите (контекстно превключване);
		- Диспечиране на процеси:
		- Координиране на взаимодействащи си процеси. Проблеми при взаимодействащите си процеси: взаимно изключване, синхронизация, комуникация;
		- Избягване и преодоляване на „Мъртва хватка”.

В случаите, когато свързани процеси работят съвместно по обща задача, възниква необходимостта от обмен на данни между тях и от синхронизация на операците им при достъп до общи ресурси. Такъв тип взаимодействие се определя като междупроцесна комуникация (Interprocess Communication - IPC) и е една от съществените задачи на ОС.

Целесъобразно е процесът да бъде определен като контейнер, в който се съхранява цялата необходима информация за изпълнение на програмата. Това включва състоянието на регистрите на процесора и брояча на командите, структури на достъпното адресно пространство, списък на отворените файлове и управляващите им блокове, списък на свързаните с основния процес други процеси и др. Всеки процес се представя пред операционата система със специфична структура от данни, наречена „блок за управление на процеса” (Process Control Block - PCB), която съдържа тези негови основни характеристики. Процесът се идентифицира със собствен уникален идентификатор (Process Identifier). Освен PID процесът ползва и специфичния идентификатор на потребителя (PID), с който той е регистриран пред системата, а също и групов идентификатор (GID), когато такъв е установен. Дъщерните процеси имат собствен PID, но наследяват PID и GID на родителския процес.

От гледна точка на изпълняваните задачи процесите могат да се класифицират на системни и потребителски. Системните процеси изпълняват код от системните модули на ОС и имат възможност да работят в привилегирован режим. Потребителски процеси са всички, които не са системни.

**7.2 Състояние на процес**

 Всеки процес се характеризира със специфично състояние, което по време на изпълнението му може да се променя динамично. Това става при иницирането и статирането на процеса, при което се създават необходимите условия и настройки (заделяне на оперативна памет, създаване на управляващи блокове и др.) за изпълнението на потребителската програма, завършване на изпълнението на програмата, при което се освобождават заеманите от програмата ресурси, освобождават се буферни области за данни и файлове. По време на същинското си изпълнение в многозадачна среда един процес също може многократно да променя състоянието си в активно (предоставен процесор и текущо изпълняван код на приложението) или в различни пасивни форми на своето състояние, свързани с изчакване на края на входно/изходни операции, получаване достъп до ресурси или изчакване в съответна приоритетна опашка за изпълнение. Промяната на състоянията на процеса е свързано с работата на управляващия компонент на ОС, наречен „диспечер на процесите” (dispatcher, scheduler), чиято цел е ефективното използване на централния процесор (респ.процесори) на КС. Основните състояния, които може да има процесът, са следните:

* + - Състояние „нов” („new”) – процесът е създаден, но програмният код още не се изпълнява;
		- Състояние „изпълняван” („running”) – процесът има достъп до процесора и неговия код се изпълнява към текущия момент;
		- Състояние „блокиран” (“blocked” или “waiting”) – кодът не се изпълнява, а процесът изчаква за настъпването на някакво външно събитие (завършване на В/И операция, получаване достъп до ресурс или др.). Следва да се има предвид, че причините за блокиране на процес са разнообразни и от там свързаните с това състояния обикновено се различават при по-детайлен анализ на този механизъм.
		- Състояние „готов” (“ready”) – процесът очаква предоставянето на процесор за изпълнение. Обикновено това е състоянието на изчакване на процеса в приоритетна опашка преди да му бъде предоставен процесор.
		- Състояние „завършил” (“terminated”) – процесът е завършил изпълнението си и по-нататъшната му работа е невъзможна.

Схема на възможните преходи между процесите, представена под формата на ориентиран граф е показана на фиг. 3 .



Фиг. 3. Граф на преходите между основните състояния при управление на процесите.

 Диспечирането на процесите е функция на ОС, при която се определя реда на предоставяне на централния процесор на съответната задача. Критериите за целесъобразно диспечиране са няколко – степен на използване на процесора, време за реакция и отговор на потребителя и т.н. При планиране на работата на диспечера на процесите под внимание се вземат такива фактори като приоритет на процеса, потребявани ресурси от процеса и др. Възможните техники на планиране и диспечиране на процесите са следните:

* + - системи с мултпрограмиране (multiprogramming);
		- системи с времеделене (time sharing);
		- системи с циклично обхождане (round robin).

Превключването на процесите от един към друг, при което те преминават в активно състояние и им се предоставя достъп до централния процесор, се нарича „контекстно превключване”. При всяко контекстно превключване в блока за управление на процеса се запазва състоянието на регистрите и програмния брояч, а от гледна точка на ОС се поддържа управляваща структура със стекова организация. Следва да се отбележи, че правиилното протичане на тези действия има изключително значение за ефективното използване на процесора (процесорите) и за реалното бързодействие на КС.

Съществена задача на планирането и диспечирането на процесите е избягването на т.нар. „мъртва хватка” (Dead Lock), при която процесите взаимно се блокират поради ползването и заемането на общи ресурси.

**7.3 Процеси и нишки**

В много случаи съвременните операционни системи допускат паралелно изпълнение на отделни участъци от кода на приложението в рамките на един процес. Изпълнимите паралелно участъци се наричат „нишки” (threads), а паралелната им работа е възможна, независимо от това дали архитектурта на КС е еднопроцесорна или многопроцесорна (фиг. 4). В първия случай паралелната работа на нишките се постига за сметка на споделянето на процесора в последователни моменти от времето чрез диспечиране на натоварването му от специален модул на ОС „диспечер на нишките” (thread dispatcher). Във втория случай в паралелното изпълнение се включват и достъпните процесори в рамките на конкретната архитектура на КС и наличните средства за управлението им. Нишките по своята същност представляват някаква форма на „олекотен” процес. Те имат собствени състояния, собствена област за разполагане на изпълнявания код и данни, за съхранение на регистрите, собствен програмен брояч, собствени управляващи блокове и динамична памет, но споделят цялото адресно пространство на процеса без ограничения и без да се изплзват специфични средства за защита и управление на достъпа. Нишките също могат да синхронизират своята работа по отношение достъпа до критични ресурси на ОС и могат да комуникират по между си. Многонишковите приложения са много популярни в случаите, когато се цели създаването на удобен динамичен графичен потребителски интерфейс или пък се осъществява многопотребителска работа до едно приложение, което изпълнява някакви сървърни функции.



Фиг. 4. Процес, съдържащ само една (главна) нишка (ляво) и многонишков процес (дясно).

1. **Концепция на паметта на КС**

Концепцията за паметта на компютърната система е сложна и често се представя като свързана в йерархична структура от няколко нива (фиг. 5).



Фиг. 5. Йерархия на концепциите за памет в КС.

Различните равнища в йерархичната концепция на паметта се различават по своите технически и функционални характеристики – капацитет, време за достъп, честота на обръщение, енергозависимост и цена на единица памет (бит). На практика йерархичната система на съхранение на данните (краткосрочно или дългосрочно) на КС се представя в йерархия от запаметяващи устройства с различни технически параметри и начин на използване. Регистрите на процесора съхраняват най-често използваните данни, използвани в оперативните изчисления. Малката и бърза хеш-памет, непосредствено свързана с процесора, работи като съхраняваща област за подмножество от често реферирани данни и инструкции от главната (оперативна) памет. Главната памет запазва кода и данните на изпълняваните процеси и също се отличава с малки времена на достъп. Паметта, разположена върху различните външни физически носители (отбелязвана като „вторична памет”) се характеризира с достатъчно продължителен срок на съхранение на данните, ниска цена, но със значително по-големи времена на достъп и невъзможност да бъде използвана непосредствено при оперативното изпълнение на програмния код.

Като най-обща и често използвана концепция тук ще бъдат отбелязани функциите на ОС по управление на главната памет.

8.1 ***Главна памет***

**Главната (оперативна) памет** е структуриран и организиран ресурс, който се използва текущо от изпълняваните в КС програми и техните данни. Както бе отбелязано в предишния раздел, всеки процес се изпълнява в собствено адресно пространство, което осигурява разполагането на програмния код, свързаните с изпълнението му данни, динамични структури и управляващи блокове. Паметта на компютърната система, използвана за кратковременно съхраняване на данни и код по време на жизнения цикъл на процеса е фундаментална концепция в структурата и функционирането на ОС. Това е ключова абстракция, която свежда фактическата организация на физическата оперативна памет до няколко прости и разбираеми модела на нейната логическа структура.

Оперативната памет може да бъде реализирана с различни технически средства, да има сложна организация и механзъм за достъп. Независимо от това най-популярният логически модел на оперативната памет е сравнително прост – това е линейния модел, при който адресното пространство е структурирано в непрекъсната съвкупност от байтове, чиито адреси започват от 0 и завършват с горната граница на съвкупно разполагаемата памет. Друга логическа организация на ОП, използвана в съвременните компютърни системи е сегментната, при която разполагаемата памет се организира в блокове (сегменти) с постоянен размер (често – 64 kB), а адресите в адресното пространство се представят чрез двойка стойности – адрес на сегмента и относително отместване спрямо началото на сегмента. Актуална и разпространена във връзка с поддържането на механизма на „виртуална памет” е и т.нар.”странична” оганизация на паметта, при която в последната се формират логически еднотипни неголеми по капацитет блокове с уникални номера. И в трите случая съществени са функциите на операционната система, при които се осигурява ефективното използване на наличното адресно пространство, споделянето му от многото едновременно изпълнявани приложения (респ.процеси) в компютърната система и осигуряване на ефикасната му защита и безопасност.

Реализирането и използването на споделената оперативна памет следва някои основни принципи:

* **Прозрачност –** едновременно в оперативната памет могат да съществуват няколко процеса без да “знаят” нищо един за друг. Всеки процес може да бъде изпълняван независимо от броя и разположението в паметта на другите процеси.
* **Безопасност (или защита) –** процесите не могат да повредят код или данни в адресното пространство на друг процес (или в това на ОС).
* **Ефективност –** степента на използване на *CPU* следва да бъде запазена, а паметта да бъде разпределена “справедливо”.
* **Преместваемост –** способността на програмата да бъде изпълнявана при различно разположение в паметта.

Фундаментална задача на уравлението на паметта (*memory management*) е да осигури сигурно (безопасно) изпълнение на програмата чрез:

* + - Разпределение на оперативната памет между текущо изпълняваните процеси в компютърната система (*sharing of memory*);
		- Изолиране и защита на оперативната памет на процеса от несанкциониран достъп (*memory protection).*

Поделянето на паметта между изпълняваните процеси изисква максимало ефективно използване на този ресурс и създаване на условия за безпроблемно изпълнение на програмния код и достъп до използваните от него данни. Стратегиите на разпределение на паметта са много, но те могат да бъдат систематизирани в две основни групи: **статично разпределение** (присъщо за по-ранните и сравнително опростени реализации на ОС) и **динамично разпределение**. Характерно за първата група стратегии е, че наличната оперативна памет се разпределя между фиксиран брой дялове – един или няколко. Размерът на фиксираните данни е постоянен, предоставеното адресно пространство на процеса е непрекъснато. Възможно е недоизползване на адресното пространство, както и недостиг на такова в случаите на процеси, изискващи по-голям ресурс памет. Преодоляването на проблема във вътрия случай се постига чрез техники на преипокриване на пространството, използвано от модулите на програмата (т.нар.оверлейни структури). Стратегиите за реализация на динамично разпределение на паметта са значително по-разнообразни. Тук се включват както системи за управление на паметтта с променлив брой дялове, с променлив размер размер на дяловете, със сегментация на адресното пространство, с фиксирани и преместваеми адреси и т.н.. Реализирането на тези стратегии осигурява значително по-ефективно използване на оперативната памет, но предполага наличието на съответни поддъжащи механизми в апаратната част на компютърната система.

Други важни функции на ОС при разпределението на паметта са:

* **Транслация на адреси** – осигурява механизъм за изобразяване на логическите и виртуални адреси в реалното физическо пространство на оперативната памет. Има отношение към преместваемостта на програмните модули в паметта и към защитата на тяхното адресно пространство.
* **Неутрализиране на фрагментацията** – фрагментацията е нежелано явление, което възниква поради наличието на неизползвани (и обикновено неизползваеми) области от паметта между адресното пространство на модулите. То значително влошава ефективността на управлението на оперативната памет.
* **Поддържане на суаппинг** (SWAP) с вторичната (дисковата) памет – един процес може да бъде временно изведен („изтласкан”) изцяло от оперативната памет под формата на „образ” в нарочно обособена област на диска, след което да бъде възстановен и работата му да продължи. Това се налага от системата за ефективно планиране на изплзването на паметта и се извършва по определени и проверени алгоритми.

Защитата на паметта може да бъде решена както с хардуерни, така и със софтуерни средства.

Съществено е да се отбележи, че всички механизми за разпределение на оперативната памет и за защита на достъпа предвиждат заделянето на специфична поостоянна област, в която се разполага резидентната част (ядрото) на операционната система, както и на транзитно зарежданите и изпълнявани нейни модули.

**8.2 *Виртуална памет***

Виртуалната памет е механизъм, който позволява на програмите да адресират памета по логически път, без да вземат под внимание фактическия ѝ размер и органзация. Замисълът на виртуалната памет е да отговори на необходимостта от едновременно разполагане и изпълнение на множество потребителски задачи конкурентно в паметта така, че тя да се използва максимално ефективно и да няма прекъсвания между изпълнението на последователните процеси. Тъй като процесите са различни по размер, превключването между един и друг процес е затруднено при конвенционалния начин на управление на паметта. За преодоляването на този проблем, виттуалната памет използва странична организация, която позволява процесите да бъдат обхванати от еднотипни блокове с фиксиран размер, наречени „страници” (pages). Програмата реферира използваната от нея памет чрез виртуални адреси, състоящи се от номер на страницата и отместване спрямо началото ѝ. Всяка от страниците може да бъде разположена на произволно място в оперативната памет. Системата на странична организация осигурява динамично съответствие между виртуалните адреси, използвани от програмите и физическите адреси на основната памет. Не е необходимо всички страници на един процес да бъдат едновремено разположени в паметта, това се изисква само за онези, чийто код се изпълнява активно към текущия момент. Един от добре известните резултати от използването на виртуална организация на паметта е, че програмите могат да бъдат изпълнявани и на машини с ограничен ресурс памет. Недостатък на подхода е известно забавяне на изпълнението поради поддържане на механизма на обмен на страници между оперативната и вторичната памет. Използването на стратегията на виртуална памет изисква съответна хардуерна и софтуерна поддръжка.

1. **Файлова система**

Файловата система е средство за организиране на вторичната памет на компютъра в логическа структура, поддържаща единството, идентификацията и манипулацията с данните чрез абстракции, а не чрез сложните технически атрибути на външните устройства. Тя е развитие на отбелязаната по-горе концепция за паметта и е една от най-видимите от потребителя части на ОС. Централната абстракция в тази логическа система е понятието „файл”, което, както е известно, обединява семантично свързана и организирана съвкупност от данни, означена с конкретно име или друг идентификатор. Абстракциите „файл” и „файлова система” са разпространени не само върху дисковите носители, а върху доста широк клас периферни устройства (SSD, флаш-памет, SD-карта и др.), предазначени за записване, съхранение и манипулиране на данни. Основната задача на файловата система е да скрие от потребителя техническите детайли на физическите носители и да му предостави удобен достъп за основните операции с данните като създаване, модифициране, четене и изтриване.

Логическата структура на съвременните файлови системи е йерархически организирана (фиг. 6). Тя позволява съхраняването и организирането на файловете в групи, в зависимост от тяхната семантика, принадлежност, режим и права на достъп, и др. Файловете, имащи отношение към работата на операционната система, се съхраняват отделно от потребителските. Логическата им организация е специфична и предопределена, а обикновено достъпът на обикновения потребител до тях в рамките на файловата система е ограничен. Всеки файл се съхранява със своите атрибути, които включват името му, неговия тип, дати на създаване и последен достъп, размер (в байтове), собственост, права и режими на достъп и др. При фактическото ползване на файла по време на изпълнението на конкретен процес, файловите атрибути се обобщават във вътрешна структура, наречена „блок за управление на файла” (File Control Block - FCB). Файловата система позволява навигация по своята структура, модификация на съдържанието си, промени на атрибутите на файловете и т.н.



Фиг. 6 . Йерархична структура на файловата система на ОС Unix.

**9.1 *Методи за разполагане на файловете***

Методите за разполагане на файловете върху физическия носител (респ.диска) имат отношение към два съществени момента:

* Ефективното използване на пространството на физическия носител;
* Средната продължителност на времето за достъп до съдържанието на файловете.

Основните методи за разполагане на файлове са следните:

* **Непрекъснато разполагане** – файловете се разполагат в последователни блокове върху физическия носител, като образуват непрекъсната област.
* **Верижно разполагане** – файловете се разполагат върху физически блокове, които не са последователни, но са свързани чрез референции във верига. Този начин на разполагане на файловете осигурява по-добро използване на паметта на физическия носител.
* **Индексирано разполагане** – използва се специфично индексиране в таблицата за разполагане на файловете (file allocation table), като целта е да се преодолеят някои проблеми (напр.външна фрагментация), възникващи при другите разпределения.

**9.2 *Управление на свободното пространство***

Свободното пространство, незаето от файловете на файловата система също подлежи на управление. За да се приложи всяка от току що описаните техники за разполагане на файловете, е необходимо да се знае кои от блоковете от дисковото пространство са свободни и достъпни. За целта наред с таблицата за разпределение на файловете (file allocation table) се добавя и контролна структура, наречена „таблица за разпределение на дисковото пространство” (disk allocation table). Тя използва различни техники за отбелязване на свободното пространство върху диска – основаващи се на битови карти или битови таблици, верижно адресиране на свободните раздели или списък на свободните блокове.

1. **Вход/изход и работа с периферийни устройства.**

Основна задача на операционната система е управлението на В/И устройства. То се осъществява от неин компонент, наречен **система за управление (супервайзор) на входа и изхода**. Тази система осигурява следните **основни функции**:

* Поддържане на информация за състоянието на устройствата.
* Управление на достъпа до устройствата.
* Разпределение и освобождаване на устройствата.
* Управление на работата на устройствата.

Системата за управление на входа и изхода управлява непосредствено работата на В/И устройствата, като потребителските програми (процеси) нямат пряк достъп до тях.

Според организацията си и начина на използване на устройствата могат да бъдат класифицирани по следния начин:

* **Блоково ориентирани** (машинно-читаеми) - информацията се съдържа във фиксирани по големина блокове. Всеки блок има свой собствен адрес върху фиюическия носител. Операциите “четене” и “запис” са независими за всеки от блоковете. Примери: дискове, ленти и др.
* **Символно ориентирани** (читаеми от потребителя) Възприемат се като поток от ASCII или Unicode кодирани символи. Приема се, че нямат блокова структура. Не е възможно адресиране и пряк достъп. Примери: принтери, клавиатура, мишка, монитор и др.

**10.1 *Начини на изпълнение на В/И операции***

* **Програмиран В/И (*Programmed I/O*)** – изпълняват се цикли на обхождане/очакване (*busy/wait cycles*) на устройствата до завършване на В/И операция (за целта се използва проверка на флаг за готовност). Недостатък на подхода е, че отнема процесорно време;
* **В/И, управляван чрез прекъсвания (*Interrupt Driven I/O*)** –процесорът инициализира В/И операция (работата на В/И устройство (чрез команда) и продължава работата си. При завършване на работата си В/И устройство генерира сигнал “прекъсване” (*interrupt*), което се обработва от нарочна процедура на ОС (*interrupt handler*).
* **Директен достъп до паметта (*Direct Memory Access*)** - осъществява обмен на данни между В/И устройство и оперативната памет без участието на централния процесор. Изисква хардуерно решение – DMA контролер.

Физическите устройства имат разнообразни и специфични технически параметри, но задача на входно-изходната система е да направи достъпът до тях удобен, унифициран и независим от техните особености. Тази задача се решава на най-ниско ниво от драйверите на устройствата.

**10.2 *Буфериране на входа и изхода***

Буферирането е техника, при която данните последователно се натрупват в специализирана област – буфер преди завършването на входната или изходна операция. След запълване на буфера, цялото негово съдържание се изпраща към получаващото устройство (за изходните операции) или към съответната област на променливите в оперативната паме (за входните операции). Този подход ускорява и опростява изпълнението на взаимодействието с входно/изходните устройства на КС поради по-рядкото обръщение към управляващите ги програми (драйвери). Буферирането е предпочитана практика при работата с периферните устройства на КС.

Като пример за физическа организация на В/И устройство е показано типичното разпределение на дисковото пространство на фиг. 7 .



Фиг. 7. Типична организация на дисковото просранство

от сектори (sectors) и писти (tracks).

Дисковото пространство е разделено на съвкупност от адресируеми блокове с еднакъв размер (обикновено 512 B), наречени „сектори”. Секторите са организирани в концентрични пътечки, наречени „писти”. Групата от концентрични писти, равноотстоящи от центъра на диска и разположени върху паралелни дискови повърхности върху обща ос се нарича „цилиндър”.

1. **Концепции на сигурността и безопасността.**

Нарастващото колективно използване на компютърни системи, наличието на споделени мрежови и Интернет-ресурси, нарастващата ценност на информацията като ресурс, свързан със социални, бизнес и политически дейности изправят използваните операционни системи пред предизивкателството за контролиране на достъпа до информацията и гарантиране на сигурността ѝ. Естеството на рисковете, свърани с безопасното и сигурно ползване на информацията варира според конкретните обстоятелства – от предпазването от случайни непреднамерени грешки, до преодоляването злонамерени агресивни интервенции с цел несанкционирано извличане или промяна на данни. Съвременните операционни системи разполагат с ефективни средства за защита и осигуряване конфиденциалността на информацията срещу различни заплахи и на различни нива. Ключовият проблем е контролиране на достъпа до компютърната система и на съхраняваната в нея информация. Основните основните средства за решаване на този проблем могат да бъдат обхванати в следните четири групи:

* **Контрол на достъпа** – осигуряване на компютърната система срещу несанкциониран външен достъп;
* **Конфиденциалност** – ограничаване на достъпа на потребителя до информация, която не е предназначена за него;
* **Интегритет на данните** – предпазване на данните от неоторизирана модификация;
* **Аутентикация** (идентификация на потребителя) – коректна проверка за идентичността на потребителя и за валидността на неговите съобщения и данни.

Системите за сигурност, прилагани в ОС позволяват прилагането на политики на достъпа, чийто основен принци е този на „минималните привилегии”, според който всеки потребител получава не повече от необходимите разрешения и права за изпълнение на неговите непосредствени задачи.

1. **Съвременни операционни системи – особености**.

Както бе отбелязано, еволюцията на операционните системи е постоянен. обективен и сравнително бързо протичащ процес. Тя е свързана както с нарастващото техническо и технологично съвършенство на хардуерните компоненти на компютърните системи, така и с разширяващите се функционалности и обхват на софтуера, а също и с непрекъснато нарастащите очаквания и изисквания на крайните потребители. Съвременните операционни системи в значителна степен отчитат мястото и присъствието на компютърните технологии като неразделна и особено важна част от съвременната икономическа, политическа и социална среда.

Основни характеристики на съвременните операционни системи:

* + - Изключително разширен, гъвкав, многофункционален и удобен обектно-ориентиран графичен потребителски интерфейс;
		- Система за регистрация на потребителите, осигуряване на индивидуална потребителска среда за изпълнение и за достъп до ресурсите на компютърната система;
		- Възможност за работа с многопроцесорна и многоядрена архитектура;
		- Поддържане на разнообразна периферия с текуща (on line) актуализация на драйвери;
		- Поддържане на разнообразни комуникации и Web-технологии;
		- Богато осигуряване с приложни програми и специализрани средства за системно обслужване, поддръжка и безопасност.



Фиг. 8. Разпределение на използваните търговски версии

на операционни системи към 2015 г .

**12.2 *Операционни системи от типа Unix/Linux***

Операционната система Unix е създадена в Bell-Laboratories в началото на 70-години на мналия век като резултат от индивидуалните изследвания на Кен Томпсън, към когото впоследствие се присъединява и Денис Ричи. Интересното е, че във връзка с разработването на тази експериментална за момента операционна система двамата ентусиасти създават популярния и особено полезен и до днес език за програмиране C. По-голямата част от Unix е написана на C като само малка част от драйверите използва машинно-ориентиран код. Първоначалната идея на Томпсън и Ричи е да съдадат малка, гъвкава и елегантна система за работа на малки компютърни системи, като за целта те ползват своя опит от разработването на предишни такива (). Операционната система Unix се радва на бърз успех най-напред в аккадемичните среди, а впоследствие и в комерсиални приложения. Системата претърпява динамично развитие в няколко ралични версии и реализации, от които най-голяма популярност получават UNIX : 4.3 BSD и System V Release 3 . Наличието на различни варианти на системата дава повод да се въведе единен стандарт за стандартизация на системите Unix от страна на IEEE, известен като POSIX (стандарт 1003.1).

 Идеите, заложени в създаването на ОС Unix, както и възникналите проблеми с унификацията ѝ и с разширението на нейните функции насочват вниманието на финландския студент Линус Торвалдс (Linus Torvalds) към разработването на собствен вариант на OS, функционално еквивалентен на Unix (1991 г.). Използван е един от известните за момента варианти на Unix – ОС MINIX.Първоначално неговата работа е свързана с пренаписването на изцяло ново ядро (монолитно) на операционната система, след което последователно от Торвалдс и мнозина други ентусиасти се съдава съвършено нова операционна система с отворен код и с практически безплатно разпространение. Системата печели огромна популярност както поради достъпността си, така и поради своята надеждност и възможност за индивидуална адаптация и модификация на програмния код.

 Въпреки множеството си варианти и реализации, операционните системи от типа на Unix и Linux имат доста сходни характеристики, по-важните от които са посочени по-долу.



Фиг. 9. Равнища на организация на ОС Linux.

**Процеси** – както и в другите операционни системи, и тук това са основните активности на ОС. Те са твърде подобни на класическите процеси на ОС. Всеки процес изпълнява една програма и поначало се състои от една нишка. Всеки процес има собствен брояч на командите и при необходимост може да създава допълнителни нишки. Linux е многозадачна система и позволява едновременната работа на няколко процеса. Потребителят може да разполага едновременно с няколко активни процеса. Освен потребителските процеси в ОС се изпълняват и няколко десетки фонови процеса, наречени „демони” (daemons), които се стартират със зареждането на ОС. Всеки процес може да създаде произволен брой подпроцеси (команда fork). Всеки процес се идентифицира с уникален идентификатор PID (Process IDentifier). Процесите могат да коминикират помежду си с някаква форма на канали и съобщения (pipes) или чрез сигнали (signals). ОС Linux притежава средства за организиране на многонишкова работа в рамките на отделен процес.

 **Памет** – управлението на паметта в ОС Linux е твърде просто с цел да може да се осъществи преместваемостта на програмите, а също – за да може да с осъществи този механизъм на различни машини с различни блокове на управление. Всеки процес в ОС Linux има адресно пространство, състоящо се от три логически сегмента: текст, данни и стек. Текстовият сегмент съдържа програмния код и е достъпен само за четене. Сегментът на данните съдържа променливи, низове, масиви и други данни, с които работи кода.Състои се от две части: инициализирани и неинициализирани данни. Съдържанието на този сегмент може да бъде модифицирано от кода. Сегментът на стека съдържа динамични структури и данни, с които работи програмата, както и някои променливи на средата. Този сегмент заема старшите адреси на адрестното пространство на процеса и запълването му става в низходящ ред – по посока на намаляване на адресите.

 **Вход/изход** – системите за В/И в Linux и Unix са прости и по принцип не се различават. По правило всички В/И устройства се интерпретират като файлове и достъпът до тях се извършва със системни обръщения read и write. В някои случаи се уточняват допълнителни параметри на устройствата. Всички В/И устройства са интегрирани във файловата система в каталога /dev и се означават със специални имена. Устройствата могат да бъдат както със символен В/И, така и с побайтов.

 **Файлова система** – файловата система на ОС Linux пртъепява няколко последователни развития от изходния си вариант, наследен от ОС MINIX. Това са системите ext и ext2, които позволяват дълги съставни имена на файловете, по-голям обем и по-добра ефективност. По принцип ОС Linux допуска работа с няколко десетки файлови системи с помощта на виртуалнта файлова система VFS (Virtual File Sysem), което се указва при компилацията на нейното ядро. Структурата и съдържанието е без значение за ОС Linux, при което не се прави разлика между двоични и текстови файлове. Файловете се групират в йерархична дървовидна структура от каталози. Основният каталог („корена”) се означава с „/”. Типични подкаталози на корена са bin – за съхранение на двоични изпълними файлове, dev – специални файлове за В/И устройства, etc – разни системни файлове, lib – библиотеки, usr – каталози на потребители.

 За повишаване на „жизнеустойчивостта” (напр.срещу внезапно спиране на захранването) Linux предоставя т.нар. „журналируеми” файлови ситеми (journaling file systems) ext3 и ext4. Основната идея на тези системи е да поддържат „журнал”, в който се следят и регистрират всички операции с файловата система.

 **Безопасност и сигурност** – Linux е разработена като многопотребителска система от самото начало, поради което средствата за безопасност и сигурност са неразделна част от нея. Всеки от потребителите на операционната система е ргистриран и се идентифицира чрез уникален идентификатор UID (User Identifier) – цяло число от 0 до 65 535. UID се използва като идентификатор на собственика при отбелязване на файлове, процеси или други ресурси. Потребителите могат да се организират в групи, които също се номерират от 16-битови цели числа чрез т.нар. GID. Потребителят може едновремено да принадлежи към няколко групи. Основният механузъм за безопасност в Linux е прост. Всеки процес носи UID и GID на своя собственик. Когато се създава файл, той получава UID и GID на своя процес. Файлът получава също и набор разрешения за досъпа, определен от създаващия го процес. Тези разрешения определят достъпа до файла за неговия собственик, членове на групата и за останалите потребители. Потребителят с UID = 0 е с особен статус на „суперпотребител” (superuser, root). Той има неограничени права за достъп до всички елементи на файловата система. Каталозите и специалните файлове (устройствата) притежават същите режими на достъп, както и обикновените файлове.

 12.2 ***Операционна система Windows***

 Операционната система Windows (Bott,2013) е една от най-разпространените комерсиални високопроизводителни операционни систем. Нейната история започва през далечната 1981 г. с появата на първата рeализация Windows 1.0, към която няма особен потребителски интерес. Популярността ѝ започва от разпространението на Windows 3.0 около 1990 г., от която за две години са продадени на 10 милиона екземпляра. Първоначално системата се изгражда и развива като графичен интерфейс на база на дисковата операционна система MS-DOS. Принципите на организация и структура на следващите варианти на Windows са поставени с реализацията Windows NT (1993 – 1996 г.). Тази реализация на системата налага концепциите за безопасност и надеждност на Windows при запазване на графичния потребителски интерфейс, които са валидни и до момента. Съвременните реализации на ОС Windows (7.0, 8.0 и 8.1) са значително усъвършенствани. Тяхната поява отбелязва настъпилите драматични промени в разпространението и в начина на използване на компютърните системи и технологии и появата на неголеми, достъпни но мощни мобилни устройства с разнообразна архитектура и конфигирация. За да се отговори на променените условия Windows 7.0 и 8.0 са построени на модулен принцип Min Win, с цел да се осигури минимално ядро на операционната система, което може дабъде разширено на различни устройства при необходимост. Целта е всяка операционна система за конкретно устройство да се организира чрез разшиение на ядрото за сметка на нови потребителски интерфейси и функции при осигуряване на възможността за възприемането ѝ от най-широк кръг потребители. Този подход е успешно приложен за устройства, които поддържат Windows Phone 8.

 Най-важните обекти на ОС Windows са процесите, нишките и сегментите. Процесите имат виртуални адресни пространства и са контейнери за ресурсите. Нишките са единици на изпълнение, които се планират от ядрото на ОС с помощта на приоритетен алгоритъм (в който винаги се изпълнява готовата нишка с най-висок приоритет, прекъсваща работата на нишките с по нисък приоритет). Сегментите представляват обекти на паметта (подобно на файловете), които могат да бъдат изобразени в адресното пространство на процеса. Образите на изпълнимите модули EXE и DLL се представят както като сегменти, така и като съвместно използваема памет.

 Windows поддържа виртуална памет със зареждане на страниците в паметта по заявка. Алгоритъмът за зареждане се основава на концепцията на работното множество. Системата поддържа няколко типа списъкъци от страници (за оптимизиране на използването на паметта). Списъците се попълват чрез специфичен механизъм, който се старае да използа повторно физическите страници, към които дълго време не е имало обръщение. Диспечирането на виртуалната памет е ориентирано към максимална производителност на входно-изходните операции.

Входът/изходът се изълняват от драйвери на устройствата, които се придържат към модела Windows Driver Model. Всеки драйвер започва с инициализация на обекта на драйвера (абстракция на реалното устройство. Обектите на драйверите са средство за представяне на реалните устройства пред системата. Те се създават от конфигурационното описание на системата или от Plug-and-Play диспечера. Устройствата се организират в стекове и пакетите от входно-изходни заявки се предават по дълбочина на сека към обслужващия драйвер. По същество входно/изходните операции са асинхронни, драйверите обикновено поставят заявките в опашка за следваща обработка и връщат управлението на извикващата страна. Томовете на файловата система (отделните физически носители – дискове, CD, SD-карти и т.н.) се интерпретират като входно/изходни устройства.

Файловата система NTFS, използвана от Windows, е оптимизирана за многопотребителска работа. Тя използва 64-битови дискови адреси и може да адресира до 264 B дисково пространство. NTFS разполага с разширени средства за устоичивост и възстановяване от грешки, безопасност, поддържане на голямо дисково просранство (RAID1 и RAID5), многопоточни файлове и др. NTFS се основава на главна таблица на файловете, която има по един запис за файл или за каталог. Всички метаданни на файловата система NTFS са част от файла. Всеки файл има множество атрибути, които могат да се съдрържат в запис главната таблица на файловете (MFT - Master File Table) или като нерезидентни – извън нея. NTFS поддържа Unicide, компресия, журнал, шифроване и много други ценни функции.

Windows има сложна система за безопасност, основаваща се на списъци за управление на достъпа и равнища на интегритета. Всеки процес има марер за аутетикация, който идентифицира потребителя и специалните привилегии, които има конкретния процес. Всеки обект има свързан с него дескриптор за безопасност, който указва собствен списък за управление на достъпа. В последните версии на Windows са добавени много функции на безопасността, в т.ч. BitBlocker за шифриране на целия том, за смесване на адресното пространство, „неизпълними” стекове, а също и други мерки, с цел да се затруднят атаки чрез препълване на буферите.

 12.3 ***Операционни системи за мобилни устройства***

Операционните системи за мобилни устройства са изключително популярни и бързо развиващи се софтуерни системи. Те реализират своите възможности в изключително динамична среда, характеризираща се с бурно развитие на елементната електронна база, архитектурата и възможностите на средствата за комуникация и за персонална работа и нарастващ потребителски интерес с разширяващи се изисквания относно тяхната функционалност. Най съществената им характеристика е неободимостта да се покрият функционално разнообразните технически устойства и разностранните изисквания на потребителите. Друга съществена особеност е интензивното им използване в ежедевната работа на потребители с различна квалификация и различно естество на решаваните задачи.

Kъм настоящия момент като най-широко разпространена и най-предпочитана операционна система за мобилни устройства с ограничена изчислителна мощност и ресурси се е наложила ОС Android Успешното ѝ развитие и доказаните ѝ предимства водят до разширяване на нейния обхват на приложимост и за кратко време тя получава популярност и върху други класове интелигентни устройства – телевизори, часовници, битови уреди и т.н. Тя е с отворен код, а исторически началото на разработването ѝ попада някъде около 2003 г. в малка калифорнийска компания Android Inc.. През август 2005 г. Android Inc. е закупена от Google с намерението да се развива дейност в областта на Web-базирани услуги за мобилни платформи.

Операционната система покрива като обхват широк диапазаон устройства (предимно мобилни) с различна архитектура и с различна изчислителна мощност и е обект на периодични обновявания и усъвършенствания. ОС Android получава общо 22 основни актуализации, последната от които към момента е LolliPop 5.1. Като такава тя позволява поддържането на потребителски интерфейс с директна манипулация – различни видове сензорни екрани на мобилни телефони и таблети.

Android е истинска отворена безплатна платформа, разработена предимно за мобилни приложения. Тя се радва на нарастваща популярност сред производителите на мобилни устройства, защото те могат да я ползват и приспособяват, без да имат никакви финансови ангажименти към съдателите ѝ. Нейно значително предимство е, че съществуването ѝ не зависи от определен доставчик, чието присъствие на пазара на специализиран софтуер може да е времено.

Android е пълноценна операционна система, която осигурява поддръжка на управлението на процесите и паметта, файлова система, драйвери за разнообразни устройства и мрежови комуникации. В основата си Android използва на ядрото на ОС Linux – също с отворен код. В ранните реализации на Android това е Linux kernel 2.6 (2.6.25 за Android 1.0), а от април 2014 г. това са основно версиите 3.4 и 3.10 на това ядро.

Android се отличава с компонентно базирана архитектура, повлияна от разнообразието в *Internet*: части от едно приложение могат да бъдат използвани от друго, при това по начин, различен от първоначално планирания. Могат да бъдат замествани дори и вградените компоненти на средата със свои собствени усъвършенствани версии. Това разширява обхвата на творческия процес в областта на мобилния софтуер

Android е многозадачна система. Тя дава възможност за едновременното изпълнение на няколко системни и потребителски процеси, както и за поддържането на множество нишки (олекотени процеси) във всеки от тях. Всеки от процесите функционира в собствено виртуално адресно пространство, като работата му е напълно изолирана от работата на другите процеси – т.е.той няма достъп до техните данни и ресурси, както и те до неговите.

Android разполага с огромно множество от вградени услуги (сървиси): услуги за локализация, използващи триангулация от *GPS* или мобилните телефонни клетки – позволяващи да се приспособят някои функции в зависимост от местоположението на потребителя, средства за работа с разнообразни сензори, пълноценна и мощна *SQL*-база от данни, позволяваща да се съчетае локалното съхраняване на данни със синхронизацията на данните в произволно свързани устройства. Всички тези вградени възможности разширяват съществено функционалността на приложенията, изпълнявани под управлението на ОС Android при намалени разходи за разработването им.

**12.4 *Архитектура на Android***

Android ((Aydin,2012), (Smith,2015), (Иванов,2016)) има характерна модулна организация и стекова архитектура. Компонентите на стековата архитектура са разпределени в четири основни слоя, всеки от които поема специфични функции, свързани с работата на операционната система и изпълняваните потребителски приложения. Това са слоевете: приложен слой, application framework, среда на изпълнение за Java-приложения и библиотеки , слой на ядрото на Android (Linux kernel). Всеки слой осигурява функционирането на слоевете над него в стека и същевременно ги изолира от детайлите, характеризиращи работата на слоевете от по-ниско ниво. Стековата организация на Android е изобразена на фиг. 10 .



Фиг. 10: Стекова архитектура на ОС Android.

Благодарение на избраната организация и архитектура на ОС Android се осигурява гъвкава и високоефективна среда за изпълнение на потребителските приложения, както и условия за тяхната преносимост. Всяко потребителско мобилно устройство, управлявано от ОС Android се превръща в многофункционален инструмент, способен да отговори на нуждите на потребителя при решаването на много ежедневни, а също професионални задачи и като средство за комуникация и поддържане на социален живот.

Благодарение на удачната си архитектура, успешните софтуерни решения, политиката на достъпност и разпространение на отворения ѝ код ОС Android печели вниманието на разработчици и потребители и справедливо се ползва от лидерски позиции на пазара на мобилни и smart-устройства Извън всяко съмнение е, че системата ще остави трайни следи и ще определи тенденциите в развитието на мобилни операционни системи за дълъг период от време.

**Заключение**

Операционната система е най-съществения софтуерен компонент на КС. Тя има решаващо значение за ефективното управление и пълноценната работа на компютърното устройство. Тя трябва да отговаря адекватно на съвременните форми и изисквания към работата с цифрови компютри, равнището на сложност, разнообразие и съвършенство на апаратната част и непрекъснато нарастващите по обем и разширяващи е по функционалност очаквания на потребителите. Доброто познаване и използване на особеностите на съвременните операционни системи е ключова предпоставка за създаване на висококачествени софтуерни приложения.

Библиография:

1. Aydin, M., Android 4: New features for Application Development, Packt Publishing, 2012, 166 pp.
2. Bott Ed, **Introducing Windows 8.1 for IT Professionals,**  Microsoft Press, 2013.
3. Cobbaut Paul, Linux Fundamentals, Интернет-ресурс: <http://linux-training.be/linuxfun.pdf>, публикуван 24.05.2015 .
4. Smith, D., Android Recipes: A Problem-Solution Approach for Android 5.0, 4rd ed., Apress©, 2015, 738 pp.
5. Silberschatz Avi, Peter Baer Galvin, Greg Gagne, Operating System Concepts, Ninth Edition, John Wiley & Sons, Inc. , 2012.
6. Silberschatz Avi, Peter Baer Galvin, Greg Gagne, Operating System Principles, Wiley – India Edition, 2006.
7. Tanenbaum Andrew S., Modern Operating Systems (3rd Edition), PEARSON EDUCATION INTERNATIONAL, 2009.
8. Николов Л., Операционни системи, Сиела, 2012.
9. Иванов М., Софтуерни приложения в среда Android, Издателство на НБУ, 2016.