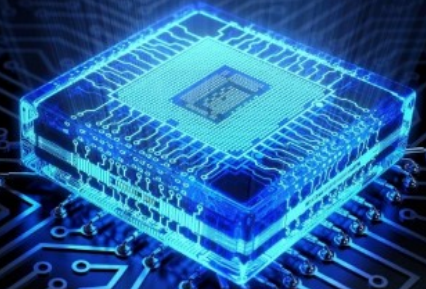
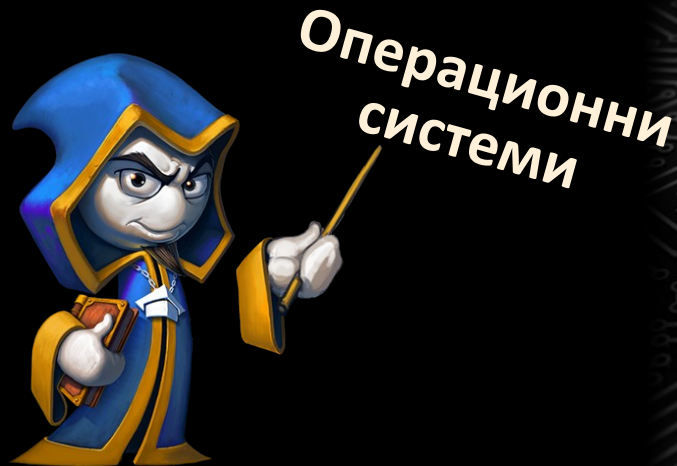


# Компютърна система

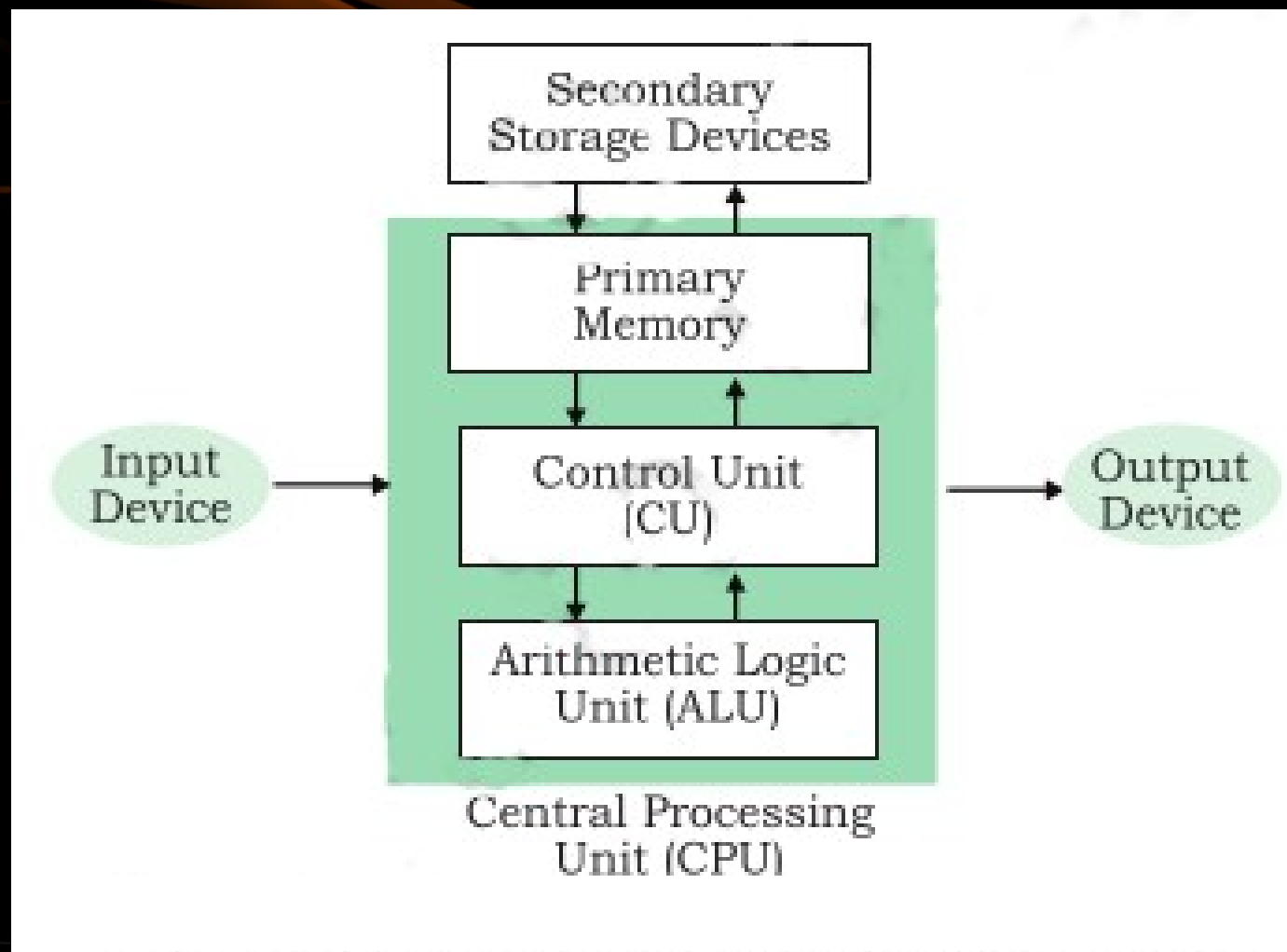


Инж. Венцеслав Кочанов

## 1.1 Въведение в компютърната система

Компютърът е електронно устройство, което може да бъде програмирано да приема данни (вход), да ги обработва и да генерира резултат (изход). Компютър заедно с допълнителен хардуер и софтуер заедно се нарича компютърна система. Компютърната система основно се състои от централен процесор (CPU), памет, входно/изходни устройства и устройства за съхранение. Всички тези компоненти функционират заедно като едно цяло, за да осигурят желания резултат. Компютърната система се предлага в различни форми и размери. Може да варира от сървър от висок клас до личен настолен компютър, лаптоп, таблетен компютър или смартфон.

Фигура 1.1 показва блоковата диаграма на компютърна система. Насочените линии представляват потока от данни и сигнал между компонентите.





## 1.1.1 Централен процесор (CPU)

Това е електронната схема на компютъра, която извършва действителната обработка и обикновено се нарича мозъкът на компютъра. Обикновено се нарича още процесор. Физически процесорът може да бъде поставен върху един или повече микрочипове, наречени интегрални схеми (IC). IC се състоят от полупроводникови материали.

На процесора се дават инструкции и данни чрез програми. След това процесорът извлича програмата и данните от паметта и извършва аритметични и логически операции според дадените инструкции и съхранява резултата обратно в паметта. Докато обработва, процесорът съхранява данните, както и инструкциите в своята локална памет, наречена регистри.

## 1.1.2 Входни устройства

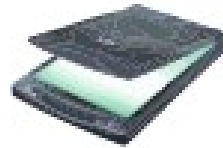
Устройствата, чрез които се изпращат управляващи сигнали към компютър, се наричат входни устройства. Тези устройства преобразуват входните данни в цифрова форма, която е приемлива за компютърната система. Някои примери за устройства за въвеждане включват клавиатура, мишка, скенер, докосване екран и т.н., както е показано на фигура 1.2.



Keyboard



Mouse



Scanner



Touch Screen

Фигура 1.2: Входни устройства

### **1.1.3 Изходни устройства**

Устройството, което получава данни от компютърна система за показване, физическо производство и т.н., се нарича изходно устройство. Той преобразува цифрова информация в разбираема за хората форма. Например монитор, проектор, слушалки, високоговорител, принтер и т.н. Някои изходни устройства са показани на Фигура 1.3.



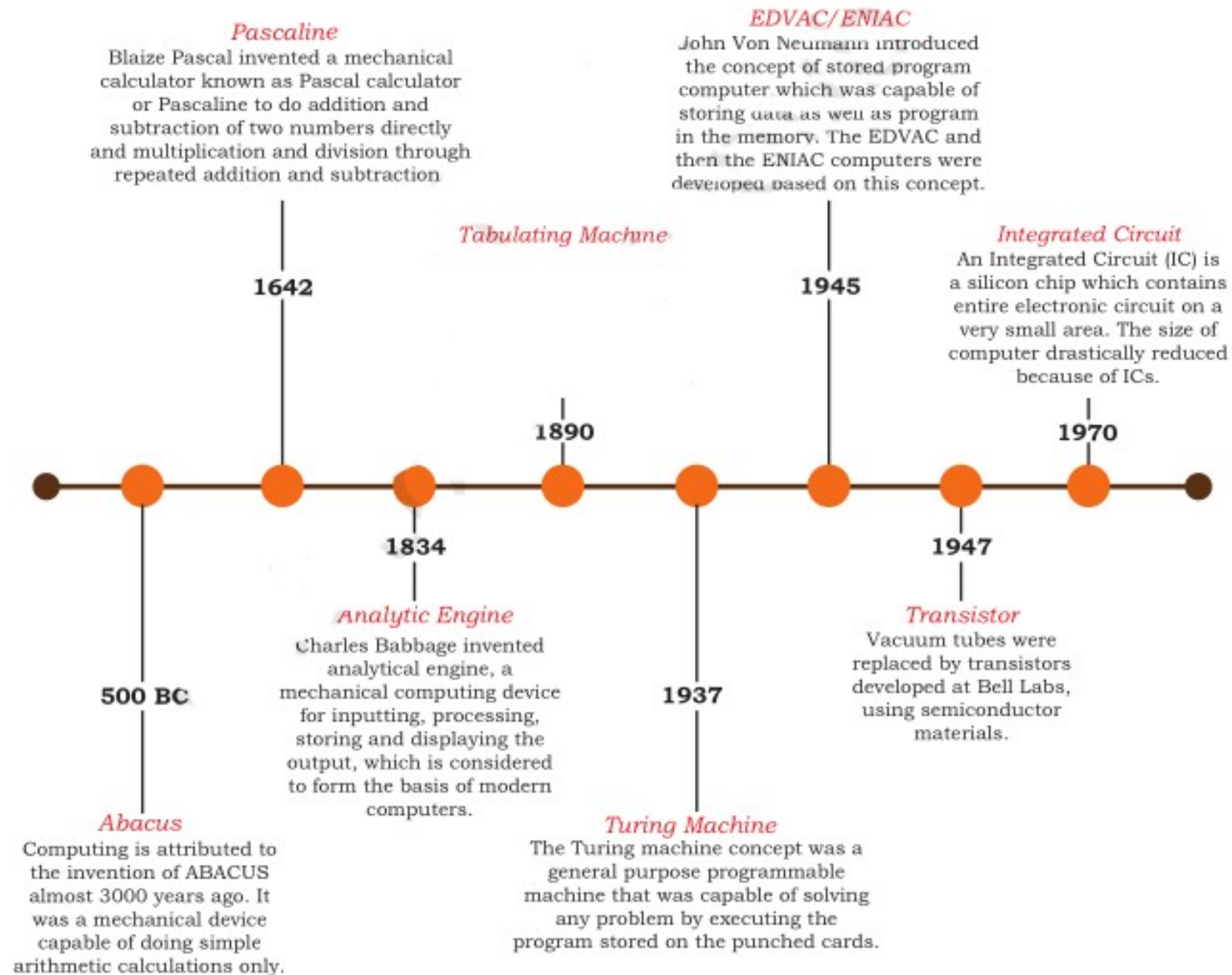


Фигура 1.3 Изходни устройства



## **1.2 Еволюция на компютъра**

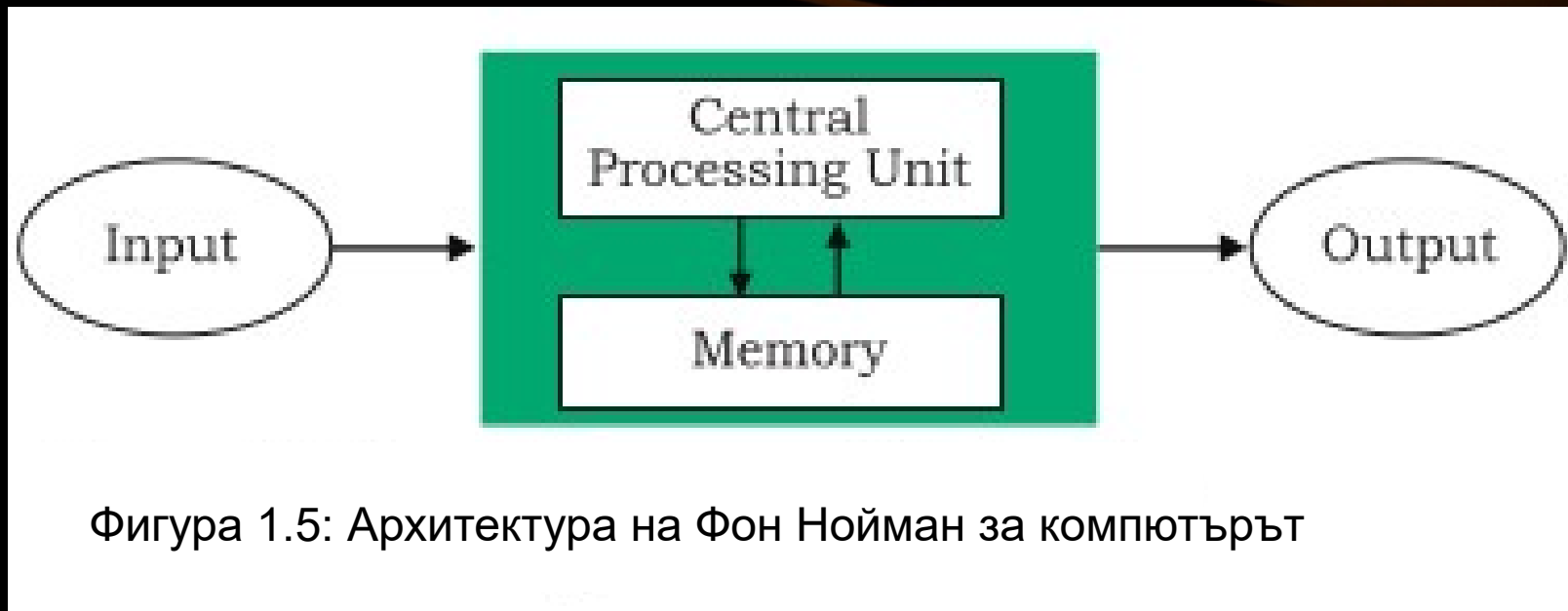
От простия калкулатор до съвременния мощен процесор за данни, изчислителните устройства са се развили за сравнително кратък период от време. Еволюцията на изчислителните устройства е показана чрез времева линия на Фигура 1.4



Фигура 1.4: Хронология, показваща ключови изобретения в компютърните технологии

**Архитектурата на фон Нойман е показана на фигура 1.5.**

Състои се от централен процесор (CPU) за обработка на аритметични и логически инструкции, памет за съхраняване на данни и програми, входни и изходни устройства и комуникация канали за изпращане или получаване на изходните данни.



Фигура 1.5: Архитектура на Фон Нойман за компютърът

### **1.3 Компютърна памет**

Компютърната система се нуждае от памет, за да съхранява данните и инструкциите за обработка. Винаги, когато говорим за „паметта“ на компютърна система, обикновено говорим за основната или първичната памет. Вторичната памет (наричана още устройство за съхранение или още твърд диск) се използва за постоянно съхраняване на данни, инструкции и резултати за бъдеща употреба.



### **1.3.2 Видове памет**

Човешките същества запомнят много неща през целия живот и извикване от паметта за вземане на решение или действие. Ние обаче не разчитаме напълно на паметта си и правим бележки и съхраняваме важни данни и информация, като използваме други носители, като бележник, ръководство, дневник, документ и т.н. По същия начин компютрите имат два вида памет — първична и вторична.

#### **(А) Първична памет**

Първичната памет е основен компонент на компютърната система. Програмата и данните се зареждат в основната памет преди обработка. Централният процесор взаимодейства директно с основната памет, за да извърши операция за четене или запис. Той е два вида, т.е. (1) Памет с произволен достъп (RAM) и (2) Памет само за четене (ROM).

RAM е непостоянна, т.е. докато компютърът се захранва, той запазва данните в него. Но веднага щом захранването бъде изключено, цялото съдържание на RAM се изтрива. Използва се за временно съхраняване на данни, докато компютърът работи. Всеки път, когато компютърът се стартира или се стартира софтуерно приложение, необходимата програма и данни се зареждат в RAM за обработка. RAM обикновено се нарича основна памет и е по-бърза от вторичната памет или устройствата за съхранение. От друга страна, ROM е енергонезависим, което означава, че съдържанието му не се губи дори когато захранването е изключено. Използва се като малко, но по-бързо постоянно хранилище за съдържанието, което рядко се променя. Например, програмата за стартиране (boot loader), която зарежда операционната система в основната памет, се съхранява в ROM.

## **(В) Кеш памет**

RAM е по-бърза от вторичната памет, но не толкова бърза, колкото компютърен процесор. Така че поради RAM може да се наложи процесорът да се забави. За да се ускорят операциите на процесора, много високоскоростна памет се поставя между процесора и основната памет, известна като кеш. То съхранява копията на данните от често достъпни първични места в паметта, като по този начин намалява средното време, необходимо за достъп до данни от първичната памет. Когато процесорът се нуждае от някакви данни, той първо проверява кеша. В случай, че изискването е изпълнено, то се чете от кеша, в противен случай се осъществява достъп до първичната памет.



## **(С) Вторична памет**

Първичната памет има ограничен капацитет за съхранение и е или летлива (RAM), или само за четене (ROM). По този начин компютърната система се нуждае от спомагателна или вторична памет за постоянно съхраняване на данните или инструкциите за бъдеща употреба. Вторичната памет е енергонезависима и има по-голям капацитет за съхранение от първичната памет. Тя е по-бавна и по-евтина от основната памет. Но не може да бъде достъпен директно от процесора. Съдържанието на вторичното хранилище трябва първо да бъде пренесено в основното памет за достъп на процесора. Примери за устройства с вторична памет включват твърд диск (HDD), CD/DVD, карта с памет и т.н., както е показано на фигура 1.7.



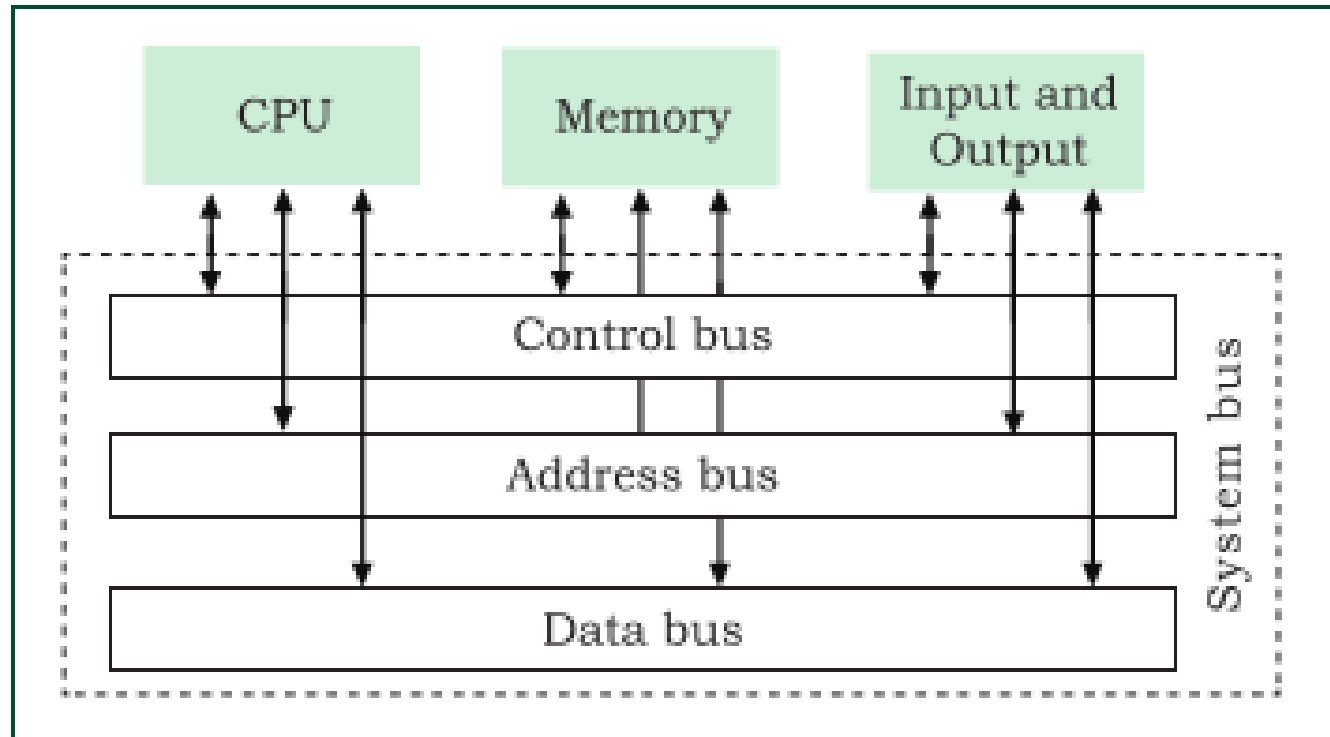


Фигура 1.7: Устройства за съхранение

## 1.4 Трансфер на данни между паметта и процесора

Данните трябва да се прехвърлят между процесора и първичната памет, както и между първичната и вторичната памет. Данните се прехвърлят между различни компоненти на компютърна система с помощта на физически проводници, наречени шина. Например шината се използва за пренос на данни между USB порт и твърд диск или между твърд диск и основна памет. Шината е от три вида: (i) **Шина за данни** за прехвърляне данни между различни компоненти, (ii) **Адресна шина** за прехвърляне на адреси между процесора и основната памет. Адресът е мястото в паметта, от което процесорът може да чете или да пише, е посочен в адресната шина и (iii) **Контролна шина** за комуникация на управляващите сигнали между различните компоненти на компютъра.

Всички тези три шини заедно правят системната шина, както е показано на фигура 1.8.



Фигура 1.8: Прехвърляне на данни между компоненти през системна шина

## 1.5 Микропроцесори

В по-ранни дни процесорът на компютъра заемаше голяма стая или множество шкафове. Въпреки това, с напредването на технологиите, физическият размер на процесора е намалял и сега е възможно да се постави процесор само на един микрочип. Процесор (CPU), който е реализиран на един микрочип, се нарича микропроцесор. В наши дни почти всички процесори са микропроцесори. Следователно, на термините се използват като синоними за практически цели. Микропроцесорът е електронен компонент с малък размер в компютър, който изпълнява различни задачи, свързани с обработката на данни, както и аритметични и логически операции.

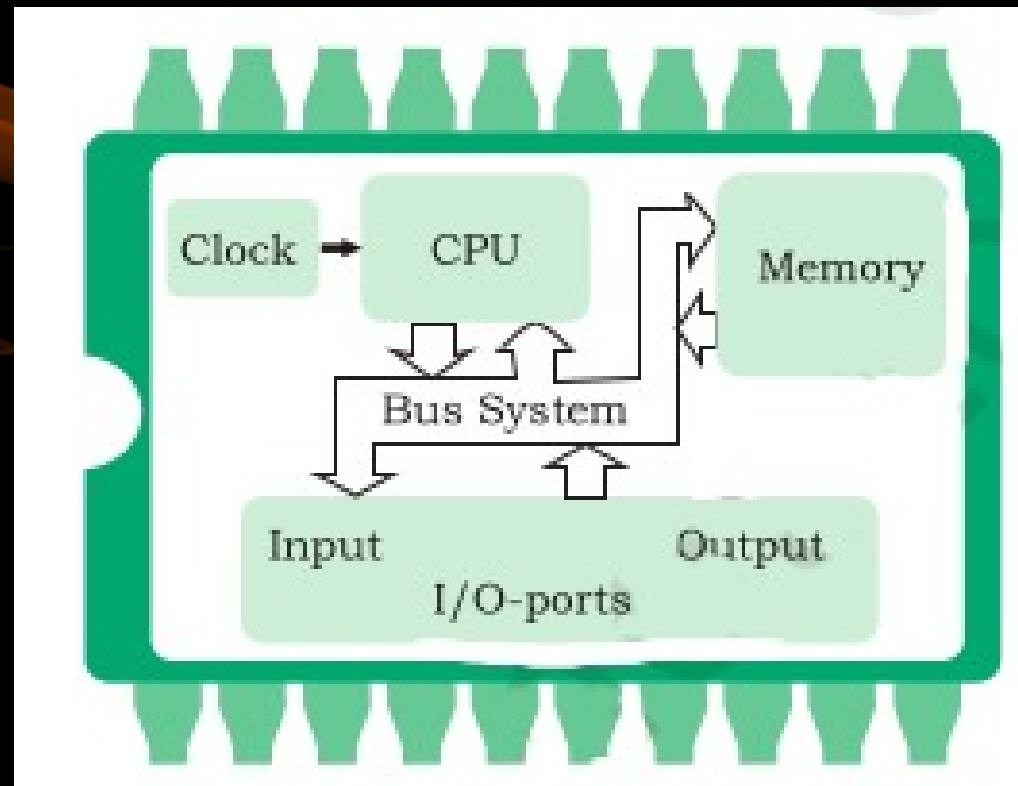


## Ядро

Ядрото е основна изчислителна единица на процесора. По-ранните процесори имаха само една изчислителна единица, като по този начин можеха да изпълняват само една задача наведнъж. С появата на многоядрения процесор стана възможно компютърът да изпълнява множество задачи, като по този начин повишаване на производителността на системата. Процесорът с две, четири и осем ядра се нарича съответно двуядрен, четириядрен и осемиядрен процесор.

## 1.5.2 Микроконтролери

Микроконтролерът е малко изчислително устройство, което има CPU, фиксирано количество RAM, ROM и други периферни устройства, всички вградени в един чип. Структурата на микроконтролера е показана на фигура 1.9. Клавиатура, мишка, цифров фотоапарат, флашка, дистанционно управление, са няколко примера за микроконтролери. Тъй като те са предназначени само за специфични задачи, следователно техният размер, както и цената им са намалени. Поради много малкия размер на микроконтролера, той е вграден в друго устройство или система, за да изпълнява специфична функция.



Фигура 1.9: Структура на микроконтролер

## 1.7 Софтуер

Досега изучавахме физическите компоненти или хардуера на компютърната система. Но хардуерът не е полезен сам по себе си. Хардуерът трябва да се управлява от набор от инструкции. Тези набори от инструкции се наричат софтуер.

Това е компонент на компютърна система, който не можем да докоснем или видим физически. Той включва инструкциите и данните, които трябва да бъдат обработени с помощта на компютърния хардуер. Компютърният софтуер и хардуер изпълняват всяка задача заедно. Софтуерът включва набор от инструкции, които при изпълнение дават желанния резултат. С други думи, всеки софтуер е написан за някаква изчислителна цел. Някои примери за софтуер включват операционни системи като Ubuntu или Windows 7/10, инструмент за текстообработка като LibreOffice или Microsoft Word ,видео плейър като VLC Player, фоторедактори като GIMP и Photoshop които рисуват.



### **1.7.1 Системен софтуер**

Софтуерът, който предоставя основната функционалност за работа с компютър чрез директно взаимодействие с неговия съставен хардуер, се нарича системен софтуер. Системният софтуер знае как да управлява и използва различни хардуерни компоненти на компютъра. Той предоставя услуги директно на крайния потребител или на друг софтуер.

Примери за системен софтуер включват операционни системи, системни помощни програми, драйвери на устройства и др.

#### **(А) Операционна система**

Както подсказва името, операционната система е системен софтуер, който управлява компютъра. Операционната система е най-основният системен софтуер, без който друг софтуер не може да работи.

## **(С) Драйвери на устройства**

Както подсказва името, целта на драйвера на устройството е да осигури правилното функциониране на определено устройство. Когато става въпрос за цялостната работа на една компютърна система, операционната система върши работата. Но всеки ден към компютърната система се добавят нови устройства и компоненти. Не е възможно само операционната система да управлява всички съществуващи и нови устройства, като всяко устройство има различни характеристики. Отговорността за цялостния контрол, работа и управление на конкретно устройство на хардуерно ниво се делегира на драйвера на устройството. Драйверът на устройството действа като интерфейс между него и операционната система. Той предоставя необходимите услуги, като скрива подробностите за операциите, извършвани на хардуерно ниво на устройството.

### 1.7.3 Инструменти за програмиране

За да свършим някаква работа с компютъра, трябва да дадем инструкции, които да се прилагат към входните данни, за да получите желанния резултат. Компютърните езици са разработени за писане на тези инструкции. Тук е важно да се разбере, че компютрите и хората разбират напълно различни езици.

Докато хората могат да пишат програми на език от високо ниво, компютрите разбират машинния език. Съществува непрекъсната нужда от преобразуване от високо ниво към език на машинно ниво, за което са необходими преводачи. Също така, за да напишете инструкцията, редактори на кодове (напр. IDLE в Python) са необходими. Тук ще опишем накратко езиците за програмиране, езиковите транслатори и инструментите за разработка на програми.



### 1.7.4 Приложен софтуер

Системният софтуер осигурява основната функционалност на компютърната система. Различните потребители обаче се нуждаят от компютърната система за различни цели в зависимост от техните изисквания. Следователно е необходима нова категория софтуер, която да отговори на различните изисквания на крайните потребители. Този специфичен софтуер, който работи върху системния софтуер, се нарича приложен софтуер. Отново има две широки категории приложен софтуер — с общо предназначение и персонализиран.



## **А) Софтуер с общо предназначение**

Приложният софтуер, е разработен за общи приложения, за да се погрижи за по-голяма аудитория като цяло. Такъв готов приложен софтуер може да се използва от крайните потребители според техните изисквания. Например, инструментът за електронни таблици Calc на LibreOffice може да се използва от всеки компютърен потребител за извършване на изчисления или за създаване на акаунт, Adobe Photoshop, GIMP, уеб браузър Mozilla, iTunes и др. попадат в категорията на софтуера с общо предназначение.

## **(В) Персонализиран софтуер**

Това са приложени софтуери, направени по поръчка, които са разработени, за да отговорят на изискванията на конкретна организация или физическо лице. Те са по-подходящи за нуждите на индивид или организация, като се има предвид, че са проектирани според специални изисквания. Някои примери за софтуер, дефиниран от потребителя, включват уебсайтове, софтуер за управление на училище, счетоводен софтуер и др. Подобно е на закупуване на парче плат и получаване на ушита по поръчка дреха с кройка, цвят и материя по наш избор.

### **1.7.5 Патентован или безплатен софтуер с отворен код**

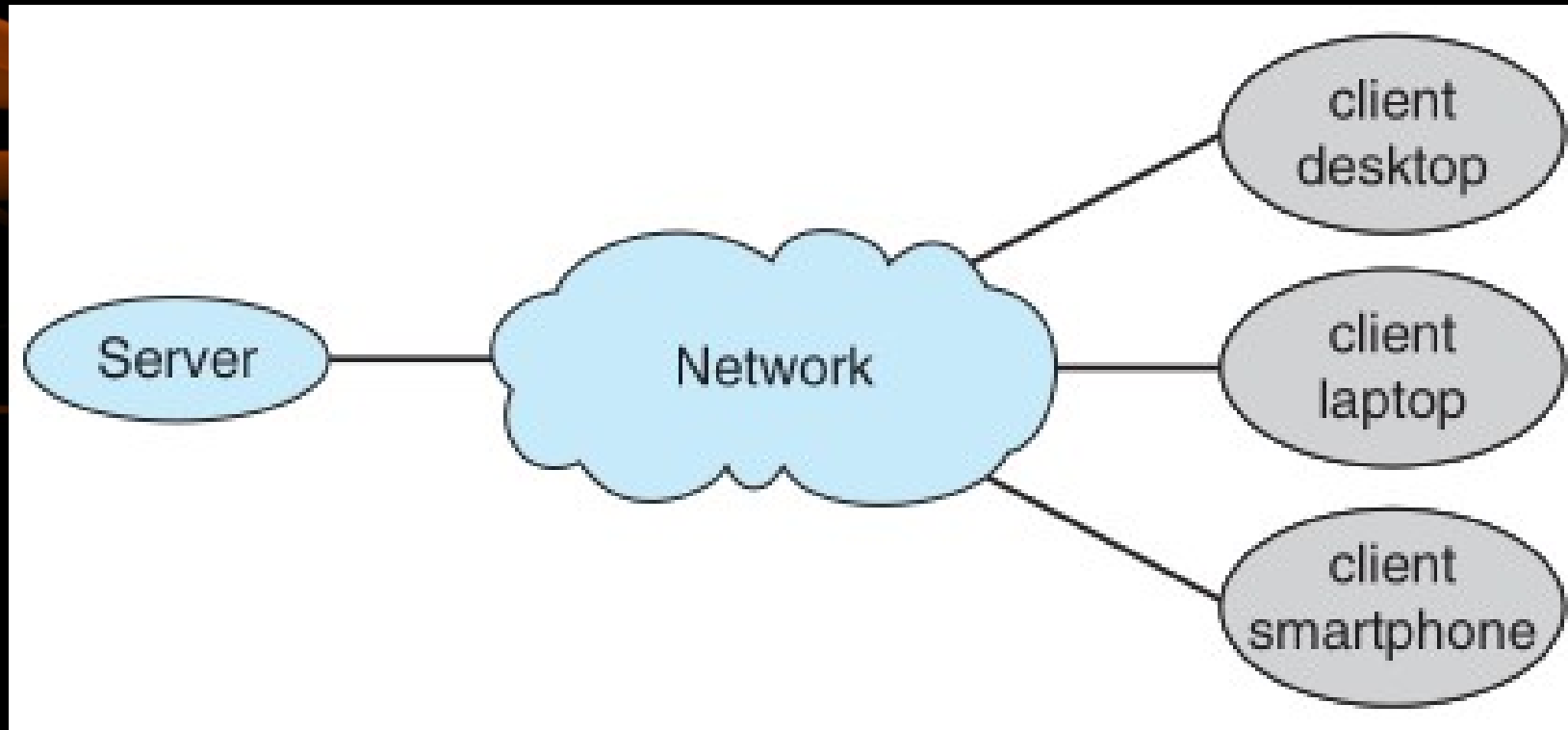
Разработчиците на някои приложен софтуер предоставят своя изходен код както и софтуера свободно за обществеността, с цел да се развиват и подобряват допълнително с взаимна помощ. Такъв софтуер е известен като безплатен и Софтуер с отворен код (FOSS). Например изходният код на операционната система Ubuntu е свободно достъпен за всеки, който има необходимите познания за подобряване или добавяне на нова функционалност. Още примери за FOSS включват Python, Libreoffice, Openoffice, Mozilla Firefox и др.



## 2. Изчисления клиент – сървър(41)

Тъй като компютрите стават по-бързи, по-мощни и по-евтини, дизайнерите се отклониха от централизираната системна архитектура. Терминалите, свързани към централизирани системи, сега се изместват от компютри и мобилни устройства. Съответно функционалността на потребителския интерфейс, управлявана някога директно от централизирани системи, все повече се обработва от компютри и доста често чрез уеб интерфейс. В резултат на това много от днешните системи действат като сървъри за задоволяване на заявки генерирани от клиентски системи. Тази форма на специализирана разпределена система, наречена клиент-сървър, има общата структура, изобразена на фигура 4.1.





Фигура 2.1 Обща структура на система клиент – сървър.

Сървърните системи могат да бъдат широко категоризирани като изчислителни сървъри и файлови сървъри:

- Системата компютърен сървър предоставя интерфейс, към който клиентът може да изпрати заявка за извършване на действие (например четене на данни). В отговор сървърът изпълнява действието и изпраща резултатите на клиента. Сървър, работещ с база данни, която отговаря на клиентски заявки, е пример за такава система.
- Системата на файловия сървър предоставя интерфейс на файловата система, където клиентите могат да създават, актуализират, четат и изтриват файлове. Пример за такава система е уеб сървър, който доставя файлове на клиенти, работещи с уеб браузъри.

### 3. Peer-to-peer компютри

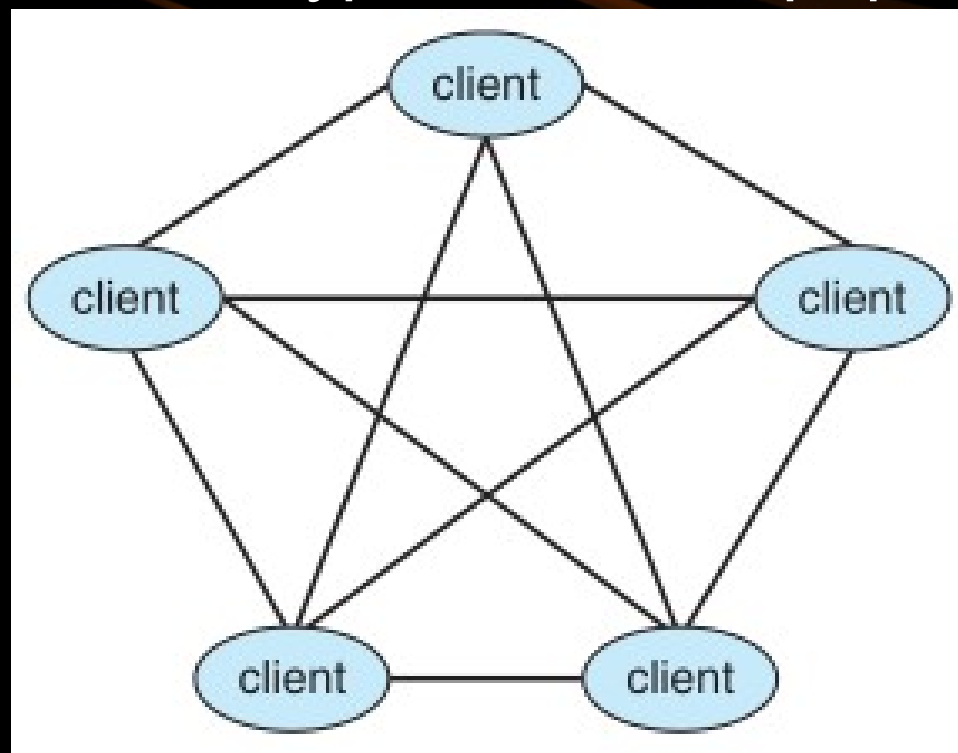
Друга структура за разпределена система е моделът на системата peer-to-peer (P2P). В този модел клиентите и сървърите не се разграничават един от друг. Вместо това всички възли в системата се считат за партньори и всеки може да действа като клиент или сървър, в зависимост от това дали иска или предоставя услуга. Системите "peer-to-peer" предлагат предимство пред традиционните системи клиент-сървър. В система клиент-сървър сървърът е тясно място; но в система peer-to-peer услугите могат да се предоставят от няколко възела, разпределени в цялата мрежа. За да участва в peer-to-peer система, възелът трябва първо да се присъедини към мрежата си от партньори.

След като даден възел се присъедини към мрежата, той може да започне да предоставя услуги на други възли и да изисква услуги от тях. Определянето на това какви услуги са налични се постига по един от двата основни начина:

- Когато възел се присъедини към мрежа, той регистрира своята услуга с централизирана за търсене в мрежата. Всеки възел, желаещ конкретна услуга, първо се свързва с тази централизирана услуга за търсене, за да определи кой възел предоставя услугата. Останалата част от комуникацията се осъществява между клиента и доставчика на услугата.
- Алтернативната схема не използва централизирана услуга за търсене. Вместо това партньор, действащ като клиент, трябва да открие кой възел предоставя желаната услуга чрез излъчване на заявка до всички други възли в мрежата.



Възелът (или възлите), предоставящ тази услуга, отговаря на партньора, който прави заявката. За да се поддържа този подход, трябва да се предостави протокол за откриване, който позволява на партньорите да откриват услуги, предоставени от други партньори в мрежата. Фигура 5.1 илюстрира такъв сценарий.



Фигура 3.1 Peer-to-peer система без централизирано обслужване.

Skype е пример за компютри peer-to-peer. Тя позволява на клиентите да извършват гласови повиквания и видео разговори и да изпращат текстови съобщения по интернет, използвайки технология, известна като глас през IP (VoIP). Skype използва хибриден подход peer-to-peer. Той включва централизиран сървър за влизане, но също така включва децентрализирани партньори и позволява на двама партньори да комуникират.

#### 4. Облачни изчисления(р.45)

Облачните изчисления са вид изчисления, които предоставят възможност за съхранение и дори приложения като услуга в мрежа. В известен смисъл това е логично продължение на виртуализацията, защото използва виртуализацията като основа за своята функционалност. Например съоръжението Amazon Elastic Compute Cloud (EC2) разполага с хиляди сървъри, милиони виртуални машини и петабайти място за съхранение достъпни за използване от всеки в Интернет.

Потребителите плащат на месец въз основа на това колко от тези ресурси използват. Всъщност има много видове облачни изчисления, включително следните:



- **Публичен облак** - облак, достъпен през интернет за всеки, който желае да плати за услугите
- **Частен облак**—облак, управляван от компания за собствена употреба
- **Хибриден облак**—облак, който включва както публични, така и частни облачни компоненти
- **Софтуер като услуга (SaaS)** - едно или повече приложения (като текстови процесори или електронни таблици), достъпни през интернет
- **Платформа като услуга (PaaS)** - софтуерен стек, готов за използване на приложения през интернет (например сървър на база данни)
- **Инфраструктура като услуга (IaaS)** - сървъри или хранилище, достъпно през интернет (например, достъпно хранилище за създаване на резервни копия на производствени данни).



## 5. Архитектура на компютърната система(р.12)

Компютърната система може да бъде организирана по няколко различни начина, които можем да категоризираме приблизително според броя на използваните процесори с общо предназначение.

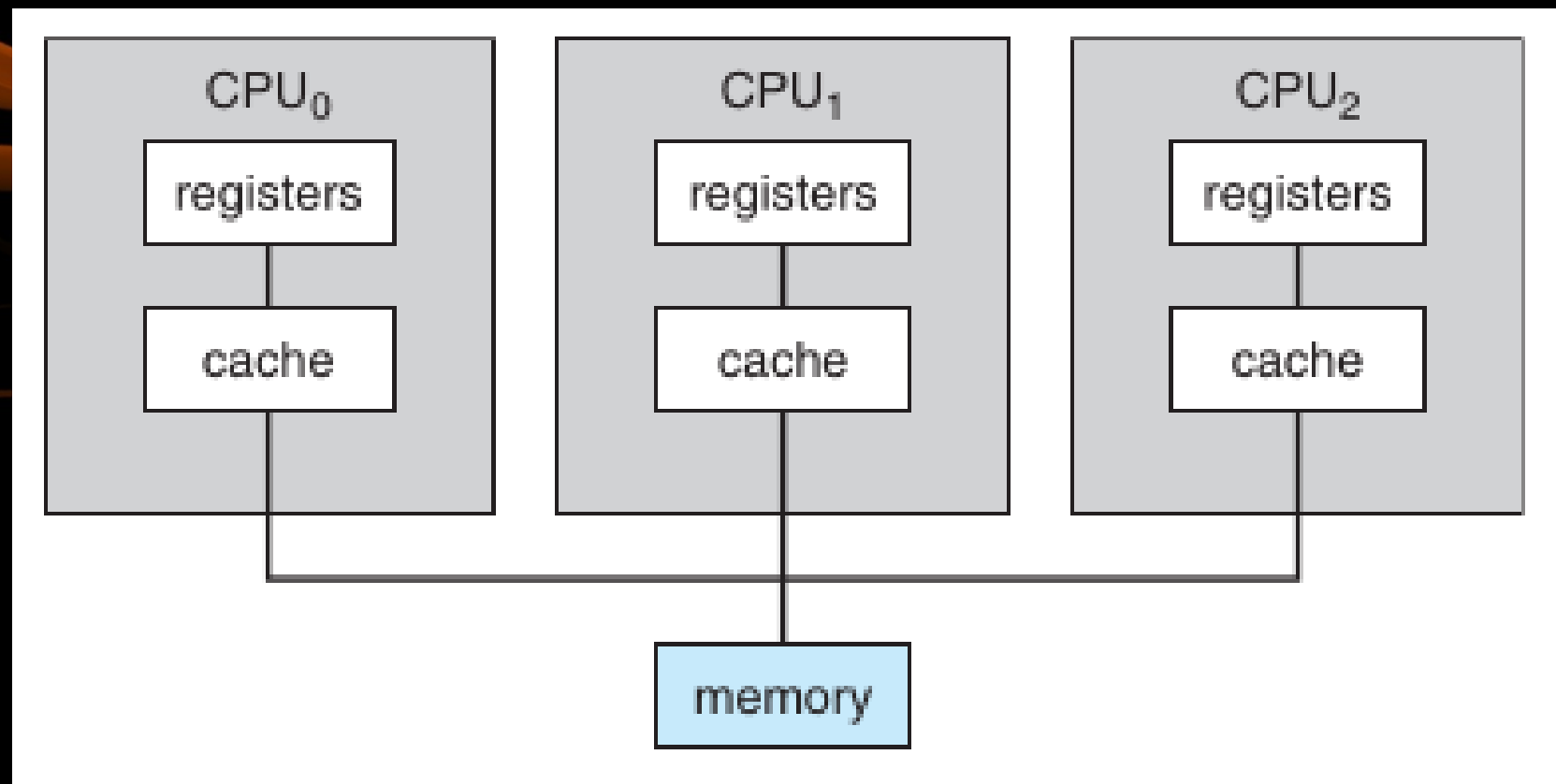
### 5.1 Еднопроцесорни системи

Доскоро повечето компютърни системи използваха един процесор. В една процесорна система има един основен процесор, способен да изпълнява набор от инструкции с общо предназначение, включително инструкции от потребителски процеси. Почти всички еднопроцесорни системи имат и други процесори със специално предназначение.

Те могат да бъдат под формата на специфични за устройството процесори, като например диск, клавиатура и графични контролери или мейнфрейми, те могат да бъдат под формата на процесори с по-общо предназначение, като I/O които преместват данни бързо между компонентите на системата. Всички тези процесори изпълняват ограничен набор от инструкции и не изпълняват потребителски процеси. Понякога те се управляват от операционната система, като тя им изпраща информация за следващата им задача и следи състоянието им. Например, микропроцесор на дисков контролер получава поредица от заявки от главния процесор и прилага своя собствена дискова опашка и алгоритъм за планиране.

## 5.2 Многопроцесорни системи(р.13)

През последните няколко години многопроцесорните системи (известни също като паралелни системи или многоядрени системи) започнаха да доминират в компютърния пейзаж. Такива системи имат два или повече процесора в тясна комуникация, споделяйки компютърната шина и понякога часовника, паметта и периферните устройства. Мултипроцесорните системи за първи път се появиха на видно място през сървъри и оттогава мигрираха към настолни и лаптоп системи. Напоследък множество процесори се появиха на мобилни устройства като смартфони и таблетни компютри. Многопроцесорните системи имат три основни предимства:



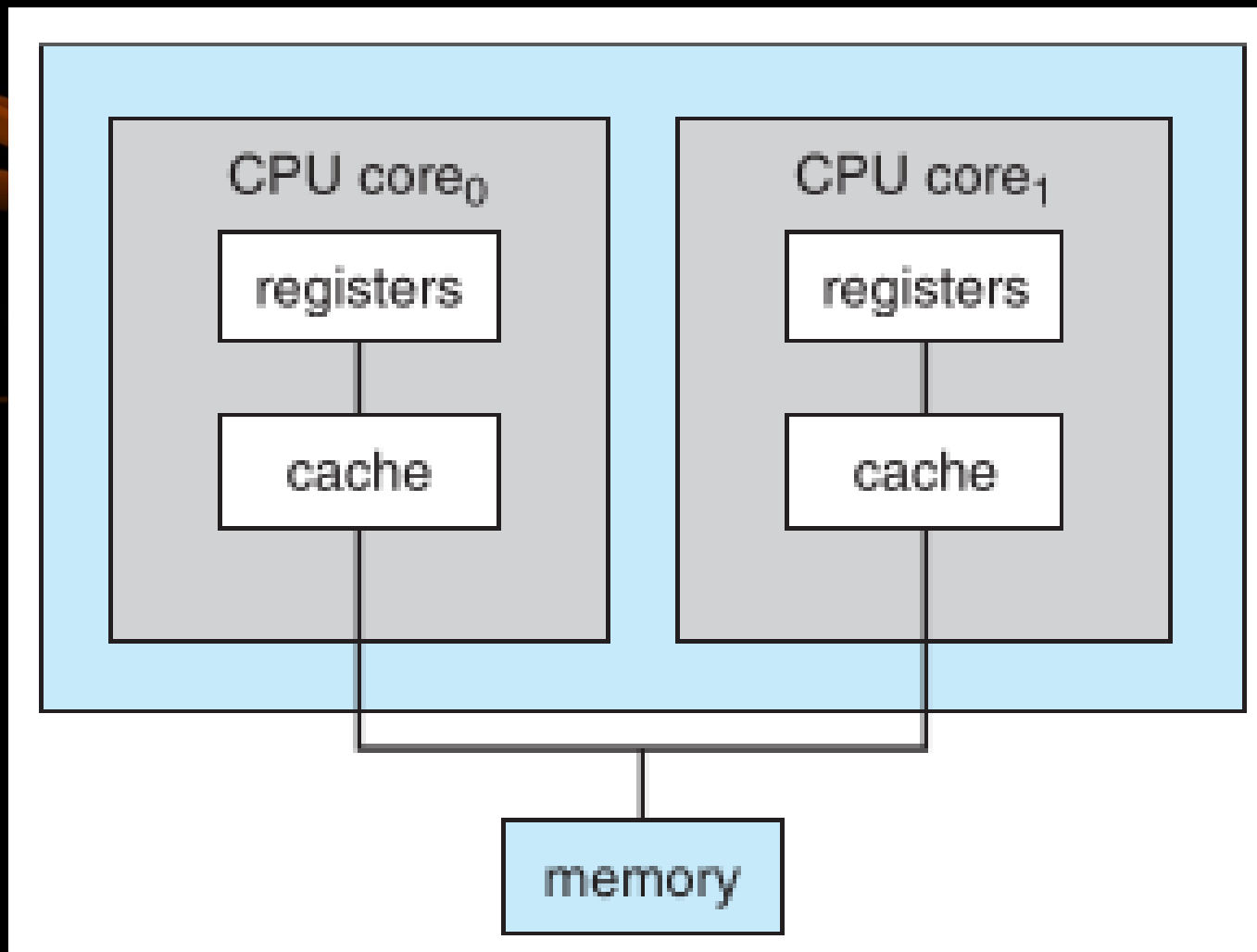
Фигура 5. Симетрична многопроцесорна архитектура.



Забележете, че всеки процесор има свой собствен набор от регистри, както и частен или локален кеш. Всички процесори обаче споделят физическа памет. На практика всички съвременни операционни системи — включително Windows, Mac OS X и Linux — вече осигуряват поддръжка за SMP.

Разликата между симетрична и асиметрична многопроцесорна обработка може да е резултат от хардуер или софтуер.

Важно е да се отбележи, че докато многоядрените системи са многопроцесорни, не всички многопроцесорни системи са многоядрени. На фигура 12.1 показваме двуюдрен дизайн с две ядра на един и същ чип.



Фигура 5.1 Двухъдрен дизайн с две ядра, поставени на един и същ чип.

## **6. Операционни системи с отворен код**

Операционните системи с отворен код са тези, налични във формат на изходен код, а не като компилиран двоичен код. Linux е най-известната операционна система с отворен код, докато Microsoft Windows е добре известен пример за обратния подход със затворен код. Операционните системи Mac OS X и iOS на Apple включват хибриден подход. Те съдържат ядро с отворен код на име Дарвин, но включват и собствени компоненти със затворен код. Започването с изходния код позволява на програмиста да произведе двоичен код, който може да бъде изпълнен на система. Правенето на обратното - обратно инженерство на изходния код от двоичните файлове е сериозна работа и полезни елементи като коментари никога не се възстановяват. Изучаването на операционни системи чрез изследване на изходния код има и други предимства.

## 6.1 Linux(49)

Като пример за операционна система с отворен код, ще разгледаме GNU/Linux. Проектът GNU създаде много UNIX-съвместими инструменти, включително компилатори, редактори и помощни програми, но никога не пусна ядро. През 1991 г. студент във Финландия, Линус Торвалдс, пуска елементарно UNIX-подобно ядро, използвайки компилаторите и инструментите на GNU и покани приноси от цял свят. Появата на Интернет означава, че всеки заинтересован може да изтегли изходния код, да го модифицира и да изпрати промени на Torvalds. Пускането на актуализации веднъж седмично позволи на тази така наречена операционна система Linux да расте бързо, подобрена от няколко хиляди програмисти. Получената операционна система GNU/Linux породя стотици уникални дистрибуции или персонализирани компилации на системата.



Основните дистрибуции включват RedHat, SUSE, Fedora, Debian, Slackware и Ubuntu. Дистрибуциите се различават по функции, помощни програми, инсталирани приложения, хардуерна поддръжка, потребителски интерфейс и цел. Например RedHat Enterprise Linux е предназначен за широко търговско използване. PCLinuxOS е LiveCD—операционна система, която може да се стартира и стартира от CD-ROM, без да се инсталира на твърдия диск на системата. Един вариант на PCLinuxOS - наречен "PCLinuxOS Supergamer DVD" - е LiveDVD, който включва графични драйвери и игри. Геймърът може да го стартира на всяка съвместима система, просто като стартира от DVD. Когато геймърът приключи, рестартирането на системата го нулира до инсталираната операционна система.

## 6.2. BSD UNIX

BSD UNIX има по-дълга и по-сложна история от Linux. Започва през 1978 г. като производно на UNIX на AT&T. Изданията от Калифорнийския университет в Бъркли (UCB) идват в изходен код и двоична форма, но не са с отворен код, тъй като се изисква лиценз от AT&T. Развитието на BSD UNIX беше забавено от съдебно дело от AT&T, но в крайна сметка през 1994 г. беше пусната напълно функционална версия с отворен код, 4.4BSD-lite. Точно както при Linux, има много дистрибуции на BSD UNIX, включително FreeBSD, NetBSD, OpenBSD и DragonflyBSD. За да разгледате изходния код на FreeBSD, просто изтеглете изображението на виртуалната машина на интересуващата ви версия и го стартирайте във VMware, както е описано по-горе за Linux.

Изходният код идва с дистрибуцията и се съхранява в `/usr/src/`. Изходният код на ядрото е в `/usr/src/sys`. Например, за да разгледаме кода за внедряване на виртуална памет в ядрото на FreeBSD, вижте файловете в `/usr/src/sys/vm`.

### 14.3. Solaris

Solaris е търговската UNIX-базирана операционна система на Sun Microsystems. Първоначално операционната система SunOS на Sun беше базирана на BSD UNIX. Sun се премести в System V UNIX на AT&T като своя база през 1991 г. През 2005 г. Sun отвори по-голямата част от кода на Solaris като проекта OpenSolaris. Закупуването на Sun от Oracle през 2009 г. обаче остави състоянието на този проект неясно. Изходният код като него беше през 2005 г. все още е достъпен чрез браузър за изходен код и за изтегляне на <http://src.opensolaris.org/source>





Въпроси?

