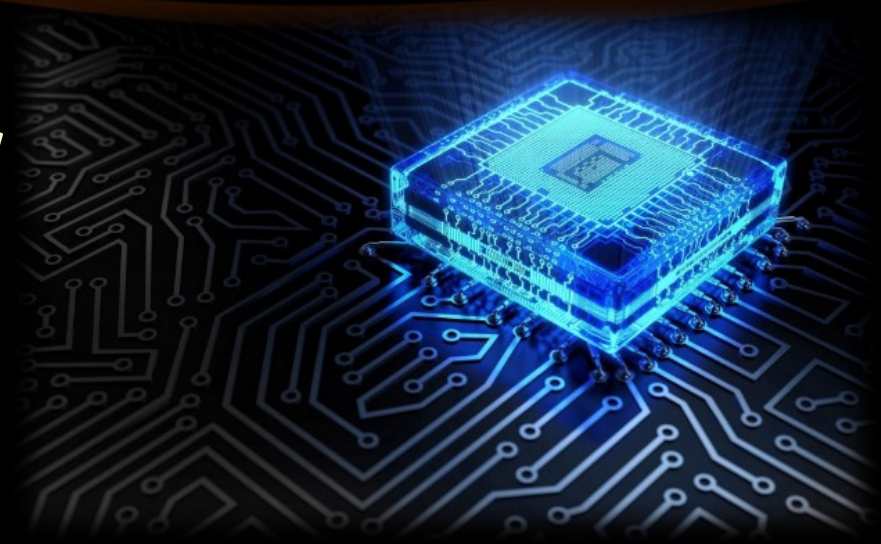
Организация на компютърната система



Инж. Венцеслав Кочанов

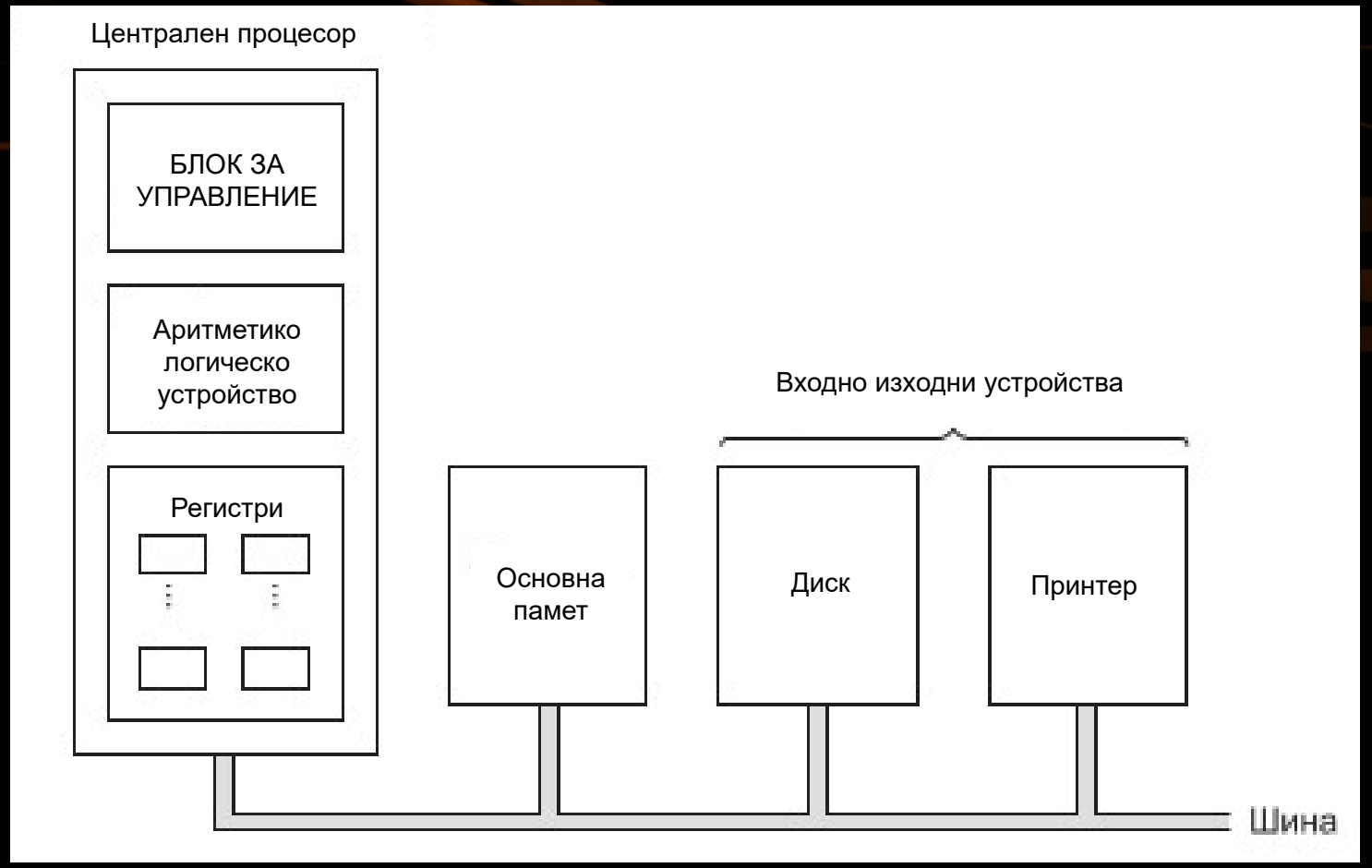
Организация на компютърните системи

Цифровият компютър се състои от взаимосвързани процесори, модули памет и входно/изходни устройства. Това са процесор, памет и входно/изходни устройства.

1.Процесор[10]

На фиг. Фигура 1.1 показва структурата на конвенционален компютър с шинна организация. Централният процесор е мозъкът на компютъра. Неговата задача е да изпълнява програми, намиращи се в основната памет. За да направи това, той извиква команди от паметта, определя техния тип и след това ги изпълнява една след друга. Компонентите са свързани с шина, която представлява набор от проводници, свързани паралелно за предаване на адреси, данни и контролни сигнали.

Шините могат да бъдат външни (свързващи процесора с памет и I/O устройства) и вътрешни. Съвременният компютър използва няколко шини.



Фиг. 1.1. Схема на компютър с един централен процесор и две входно-изходни устройства

Процесорът се състои от няколко части. ***Блокът за управление*** е отговорен за извикването на команди от паметта и определянето на техния тип. ***Аритметичното логическо устройство*** изпълнява аритметични операции (като събиране) и логически операции (като логическо И).

Регистрите са специални клетки на паметта, физически разположени вътре в процесора. За разлика от RAM, която изисква адресна шина за достъп до данни, процесорът има директен достъп до регистрите. Това значително ускорява работата с данни.

Вътре в централния процесор има малка, бърза памет наречена ***кеш памет*** която се използва за съхраняване на междинни резултати и някои контролни команди. Тази памет се състои от няколко регистъра, всеки от които изпълнява определена функция. Обикновено размерът на всички регистри е еднакъв.

4. Обиколка на компютърните системи

Компютърната система е съвкупност от хардуерни и софтуерни компоненти, които работят заедно, за да изпълняват компютърни програми. Конкретните реализации на системите се променят с времето, но основните концепции не.
Всички системи имат подобни хардуерни и софтуерни компоненти, които изпълняват подобни функции.
В техния класически текст на почти всички езици за програмиране използваме програмата hello world, като в нашият случай ще използвам езика C. Кодът е показан по-долу:

```
1 #include <stdio.h>
2
3 int main()
4 {
5         printf("hello, world\n");
          return 0;
6 }
```

Въпреки че hello е много проста програма, всяка голяма част от системата трябва да работи съгласувано, за да може да работи докрай. Целта ни е тук да разбереме какво се случва и защо, когато стартирате hello на нашата система. Ще започнем нашето изследване на системите, като проследим живота на програмата hello от момента, в който е създадена, отпечата простото си съобщение и се прекрати. Докато проследяваме жизнения цикъл на програмата, ще представим накратко основните концепции, терминология и компоненти, които влизат в действие.

4.1. Програмите се превеждат от други програми в различни форми

Програмата hello започва живота си като С програма от високо ниво, защото може да бъде прочетена и разбрана от хората в тази форма. Въпреки това, за да стартирате hello.c в системата, отделните С изрази трябва да бъдат преведени от други програми в последователност от инструкции на машинен език на ниско ниво. След това тези инструкции се пакетират във форма, наречена изпълнима обектна програма, и се съхраняват като двоичен дисков файл. Обектните програми също се наричат изпълними обектни файлове.

В Unix система преводът от изходен файл към обектен файл се извършва от драйвер на компилатор:

linux> gcc -o hello hello.c

Тук драйверът на компилатора на GCC чете изходния файл hello.c и го превежда в изпълним обектен файл hello. Програмите, които изпълняват четирите фази са: (препроцесор, компилатор, асемблер и линкер), са известни като система за компилиране.

Какво е Intermediate Language (ILDASM & ILASM) Code в C#?

Междинният език (IL) е обектно-ориентиран език за програмиране, предназначен да се използва от компилатори за .NET Framework преди статично или динамично компилиране към машинен код. IL се използва от .NET Framework за генериране на машинно-независим код като резултат от компилация на изходния код, написан на който и да е .NET език за програмиране. ILDASM означава Intermediate Language Disassembler, а ILASM означава Intermediate Language Assembler.

Йерархия на устройствата за съхранение

Идеята за вмъкване на по-малко, по-бързо устройство за съхранение (напр. кеш памет) между процесора и по-голямо, по-бавно устройство (напр. основна памет) се оказва добра идея. Основната идея на йерархията на паметта е, че съхранението на едно ниво служи като кеш за съхранение на следващото по-ниско ниво.

Цели на операционната система :

(1) да защитава хардуера от злоупотреба от неработещи приложения

(2) да предоставя на приложенията прости и единни механизми за манипулиране на сложни и често изключително различни хардуерни устройства от ниско ниво.

RISC и CISC технологии

RISC и CISC са два различни вида компютърни архитектури, които се използват за проектиране на микропроцесорите, които се намират в компютрите. Основната разлика между RISC и CISC е, че RISC (компютър с намален набор от инструкции) включва прости инструкции и отнема един цикъл, докато CISC (компютър с комплексен набор от инструкции) включва сложни инструкции и отнема множество цикли.

Прекъсвания

Какво е прекъсване?
Прекъсването е сигнал, излъчван от устройство, свързано към компютър, или от програма в компютъра. Това изисква операционната система (ОС) да спре и да разбере какво да прави по-нататък. Прекъсването временно спира или прекратява услуга или текущ процес.

Видове прекъсвания

I. Хардуерно прекъсване

Хардуерното прекъсване е електронен сигнал от външно хардуерно устройство, което показва, че се нуждае от внимание от страна на операционната система.

II. Софтуерни прекъсвания

Софтуерно прекъсване възниква, когато приложна програма прекрати или поиска определени услуги от операционната система.

Въпроси?