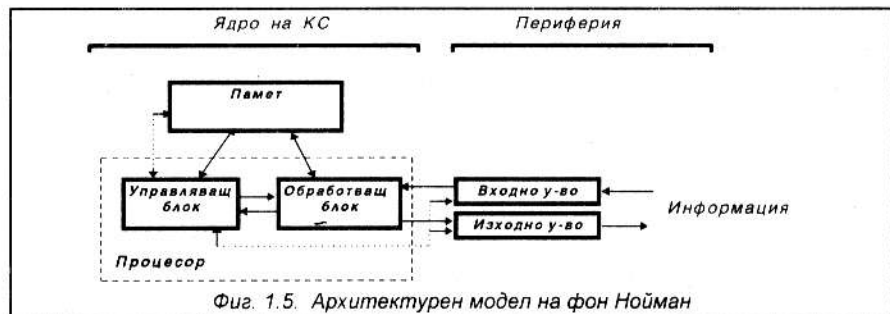


1.3. Компютърна архитектура и организация на обработката

1.3.1. Понятие за компютърна архитектура

Класическият архитектурен модел на фон Нойман (фиг.1.5), който дефинира структурата на една КС, е първата модулна концепция за структурна организация и включва самостоятелни компоненти, работещи съвместно: централна част (ядро); външна част (периферия) и функционален алгоритъм (програмно осигуряване).

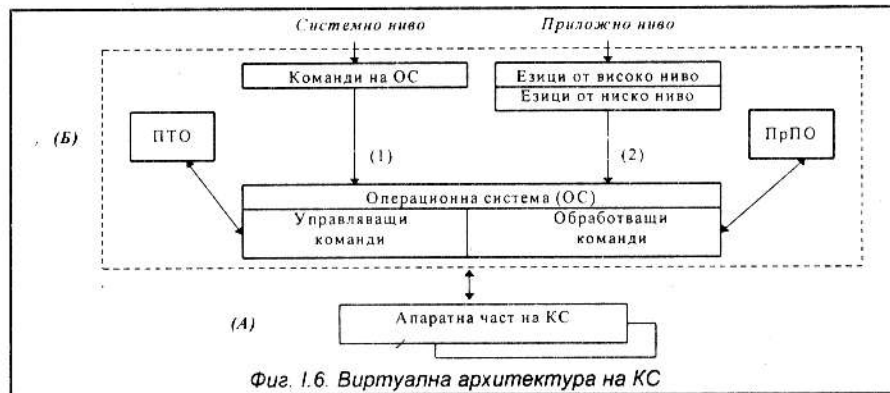


Фиг. 1.5. Архитектурен модел на фон Нойман

Организацията на информационната обработка се основава на принципа на програмното управление и разделяне на управлението от непосредственото изпълнение на операциите в операционния блок. За целта се поддържа адресен достъп до паметта за инструкции и данни, като последните са представени в унифицирани формати. Механизмът за обработката е последователно изпълнение на операциите, зададени чрез включените в дадена програма команди. Така се дефинира изчислителен процес, който се развива във времето и натоварва отделните устройства от КС с определени задачи. В този смисъл компютърната обработка е непрекъснато взаимодействие между двете основни страни на КС: (А) - апаратните средства; (Б) - програмното осигуряване и общата организация на работата, които дефинират т.нар. **виртуална архитектура** на КС (фиг.1.6). Развиват се два типа процеси - системни (1) и приложни (2), които работят с отделните компоненти на ОС, програмите за техническо обслужване (ПТО) и приложното ПО (ПрПО).

Като цяло ПО обединява всички програмни компоненти, които позволяват на потребителя да използва възможностите на апаратната част на КС. В зависимост от предназначението са обособени две части: системна (за управление на ресурсите в КС) и приложна (за разработване и реализация на компютърни приложения). Основен елемент на системната част е ОС, която осигурява режима за работа на КС. Приложната част

предоставя възможност за използване на пакети от програми или специализирани програмни системи при решаване на потребителските задачи.



Фиг. 1.6. Виртуална архитектура на КС

1.3.2. Особенности и абстрактно представяне на компютърната обработка

Профил на процесите

Компютърната обработка представлява организирана съвкупност от процеси, свързани с обслужване на потребителските задачи в КС. Един процес се описва чрез тройката $I = \langle t, A, T \rangle$, където: t - начален момент на процеса; A - дефиниращи атрибути, които определят източника на процеса (напр. потребител, програма и пр.); T - трасе на процеса, характеризиращо последователността от събития по заемане на елементи от системния ресурс S . Дадено събитие e , се характеризира с: момент на възникване; име на заеман системен ресурс; параметри за използваемостта на ресурса.

Процесите са два вида: (а) приложни процеси - за обслужване на потребителя чрез изпълнение на една потребителска задача; (б) системни процеси - пораждат се от операционната система за поддържане на приложни процеси и зависят от параметрите на отделна задача, дефинираща приложен процес, както и за обслужване на възникнали системни събития в КС, породени от вътрешни или външни причини (прекъсване, грешка, обмен, текущ контрол).

Профил на изчислителен процес е последователността от фази, реализиращи отделни събития за изграждане на цялостен изчислителен процес в КС. Примерен профил за една хипотетична ситуация е представен на фиг.1.7. Събитията e_i могат да бъдат: четене на данни; чакане във

входна опашка за планиране; заемане на системен ресурс; процесорна обработка; работа с външна памет; обмен на данни и пр.



Фиг. 1.7. Примерен профил на процес в КС

Поддържането на различни процеси при организация на компютърната обработка води до натоварване на отделните системни компоненти S_j , което може да се представи чрез т.нар. профил на системно натоварване. Дефинира се чрез вектор $V(t) = \langle v_1, v_2, \dots, v_n \rangle$, чиито елементи отразяват свободно ($v_j=0$) или заето ($v_j=1$) устройство S_j ($j=1, 2, \dots, n$) в последователни моменти от време $t=1, 2, \dots$.

Системни характеристики

Системните характеристики отразяват организацията на компютърната обработка и поведението на КС при изпълнение на потребителски задачи. Някои от основните характеристики са следните:

♦ **Работно натоварване** - заетостта на устройство S_j през интервалите τ_{ij} ($i=1, 2, \dots, k$) за период от време T :

$$R_j = \frac{1}{T} \sum_{i=1}^k \tau_{ij} \leq 1.$$

При многозадачна работа в КС се развиват няколко процеса, които натоварват различни устройства (паралелна заетост) и тогава се говори за системно натоварване:

$$R = \sum_{j=1}^n R_j.$$

♦ **Време за отговор** - времето u_i от постъпване на текуща задача за обработка до получаване на резултата. Най-често се определя т.нар. средно време за отговор за N задачи, характеризиращо функционалните способности и бързодействието на КС:

$$u_{av} = \frac{1}{N} \sum_{i=1}^N u_i.$$

♦ **Надеждност и отказоустойчивост** - определят способността КС да изпълнява коректно функциите си за определен период от време. Оценява се чрез интензивността на отказите (брой откази за единица време).

♦ **Стойност (цена)** - сумарна оценка за стойността на техническите средства и програмното осигуряване при решаване на една задача:

$$C = \sum_{i=1}^n k_i \cdot W_i,$$

където W_i е използван обем от i -тия системен ресурс при решаване на задачата, а k_i е стойностен коефициент, определящ теглото на ресурса при изчисляване на стойността на обработката.

♦ **Множествен достъп до ресурса** - дефинира способностите на КС да управлява множество от процеси, които се опитват да заемат едновременно един или няколко ресурси. Обикновено се поддържа диспечер, който поема функциите по разпределянето на системния ресурс между паралелните процеси и задава правилата за достъп и защита на общите обекти (данни, програми, устройства и пр.).

♦ **Ниво на паралелизъм** - способност на ОС да обработва паралелно различни по обем информационни единици. В зависимост от това могат да се дефинират локален и глобален паралелизъм, като и ниско и високо ниво на паралелизъм при компютърната обработка.

♦ **Стратегия (алгоритъм) за планиране на задачите** - механизъм за дипечериране на независими процеси в компютъра, който въвежда правила при обслужване на постъпващите за обработка задачи.

Производителност и ефективност

Производителността на КС е обобщено понятие, изразяващо извършената изчислителна работа за единица време:

$$P = \frac{Work}{Time}.$$

Получаване на директна количествена оценка по тази формула, обаче, е затруднено от невъзможността точно и ясно да се дефинира понятието "изчислителна работа". Тази е причината да се определят оценки за т.нар. индекси на производителността, като например оценка на броя задачи, решавани от КС за единица време:

$$\lambda = \frac{Number\ of\ processed\ tasks}{Time}.$$

Индексите на производителността на КС се измерват с различни по дименсия единици, а общата производителност е сложна нелинейна функция от тях. Примерни индекси на производителността са: брой инструкции, изпълнявани от процесора за единица време; скорост на информационния обмен между отделни устройства; цикъл на оперативната памет; пропускателна способност на информационните магистрали и пр.

На базата на производителността на КС се дефинира друг комплексен показател на функционирането - **ефективност на КС**. Той определя доколко е целесъобразно прилагането на КС за решаване на даден клас от

задачи при оценяване на стойността и разходите за организацията на изчисленията. Тук се включват и разходите, свързани с експлоатацията на КС, поддържането на нейната работоспособност, проверката на техническите показатели и разходите по отстраняване на неизправностите.

Последното е свързано с *надеждността на КС* - свойството системата да запазва своите експлоатационни показатели в определени от производителя граници. Надеждността на невъзстановимите устройства често се измерва с вероятността за безотказна работа $p(t)=[N_0-n(t)]/N_0$, където N_0 е началният брой произведени устройства, а $n(t)$ е брой на отказалите устройства за интервала от време t .

Друга характеристика е *вероятността за отказ*: $q(t)=1-p(t)=n(t)/N_0$. Надеждността на възстановимите устройства обикновено се изчислява като средна честота на отказите - среден брой на откази на едно устройство за единица време в определен интервал.

Абстрактно представяне на компютърната обработка

Абстрактното представяне на системния ресурс и компютърната обработка позволява по-ясно описание и визуализация на структурата и взаимодействието между компонентите. За целта могат да бъдат използвани различни средства, ориентирани към специфичните особености на КС и процеси. В този смисъл, подходящ апарат за абстрактно представяне са *графите*, които лесно могат да бъдат описани и изследвани програмно. Поради стохастичния характер на процесите в КС, обаче, е препоръчително използване на методи и средства с вероятностен характер (марковски и полумарковски процеси, масово обслужване и пр.)

Графът е съвкупност от върхове и ребра, означена чрез $G(V,A)$, където $V=\{V_1, \dots, V_n\}$ е множеството на върховете, а $A=\{a_{ij}\}$ е множеството от всички ребра, свързващи различни двойки върхове $\langle V_i, V_j \rangle$. Даден граф може да бъде:

- неориентиран (ненасочен), ако за ребрата няма дефинирана посока (т.е. $a_{ij}=a_{ji}$);
- ориентиран (насочен), при който за всяко ребро са дефинирани логически "начало" и "край" (тогава $a_{ij} \neq a_{ji}$);
- тегловен - ако за всяко ребро е дефинирана стойност (тегло).

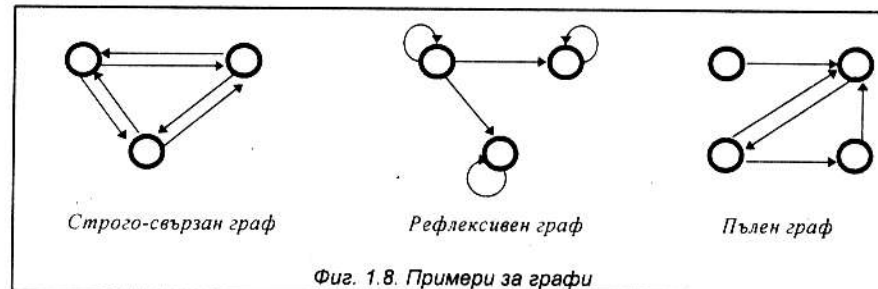
Някои основни понятия и определения при графите са следните:

- съседни върхове - върхове, свързани с общо ребро;
- входящи и изходящи ребра - влизащите $\{a_{ki}\}$ и излизащите $\{a_{iq}\}$ за връх V_i , ребра;
- път в граф - последователност от ребра, позволяващи свързване на несъседни върхове; прост път е този, който преминава еднократно през последователните върхове; цикъл - път с един и същ начален и краен връх;

г) *пълно-свързан граф* - ненасочен граф с ребра между всяка двойка върхове; *строго-свързан граф* - насочен граф, при който за всяка двойка върхове $\langle V_i, V_j \rangle$ съществуват ребрата a_{ij} и a_{ji} (двата типа са идентични);

д) *рефлексивен граф* е граф, съставен само от рефлексивни върхове (върхове с изходяща дъга, която е и входяща, т.е. за $\forall V_i, i=1, 2, \dots, n \Rightarrow \exists a_{ii}$);

е) *пълен граф* - граф, при който за всяка двойка върхове съществува поне един насочен път в една от двете посоки.



Връх V_j е достижим от връх V_i в даден граф, ако съществува насочен път $\langle V_i \rightarrow V_j \rangle$. Ако съществува и обратен път, то връх V_j е контрадостижим от връх V_i . С достижимостта в графите са свързани следните две матрици:

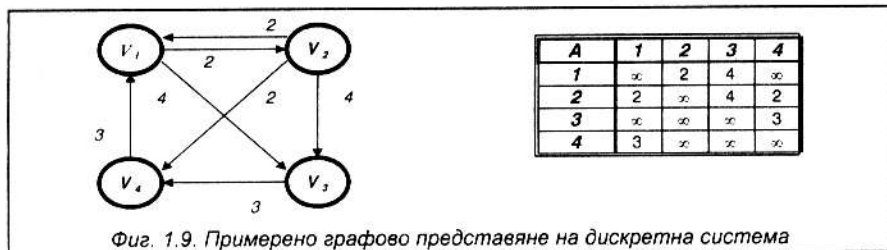
а) *Матрица на принадлежност (МП)* - двумерна структура от данни, елементите на която маркират наличието (1) или не (0) на дъга между всяка двойка възли;

б) *Матрица на достижимост (МД)* - двумерна структура от данни, елементите на която маркират наличието (1) или не (0) на път в съответния граф за всяка двойка възли.

За тегловен граф бинарните матрици се преобразуват в числови, като елементите показват теглата по съответните ребра в графа (за МП) и пътищата с минимална дължина за всяка двойка възли (за МД). Обикновено отсъствие на ребро или път се маркира с ∞ .

От функционална гледна точка една КС може да се представи като крайно множество $S=\{s_1, s_2, \dots, s_n\}$ от състояния на компютърната обработка. Тази концепция позволява за *детерминирано* абстрактно представяне да се използва тегловен граф, задаван чрез матрица на връзките (матрица на принадлежност) между основните състояния на описвания обект. Матрицата на принадлежност може да бъде булева (наличие или отсъствие на връзка) или числова (дължина на връзката, закъснение при прехода, брой канали за връзка, вероятност за преход и пр.).

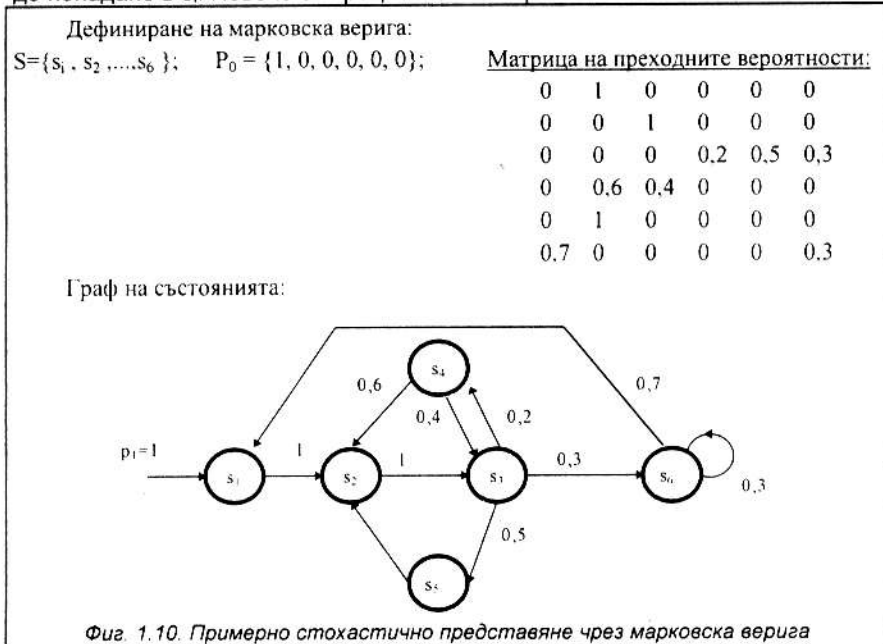
На фиг. 1.9 е представен примерен абстрактен модел на мрежова среда с 4 работни станции или терминали и закъснение при предаване на информацията по виртуални или друг тип канали, представено чрез ребрата на матрицата А.



При стохастично абстрактно представяне може да се използва **марковска верига**, която се дефинира еднозначно чрез:

- множество на възможните състояния $S = \{s_1, s_2, \dots, s_n\}$;
- матрица на вероятностите за преход $P = \{p_{ij}\}$ между състоянията $s_i \rightarrow s_j$ ($i, j = 1, 2, \dots, n$);
- вектор на началните вероятности $P_0 = \{p_i / i = 1, 2, \dots, n\}$.

Марковската верига е случаен процес, описващ поведението на една вероятностна система, в която преходите $s_i \rightarrow s_j$ от едно състояние към друго зависят само от изходното състояние s_i (от съответните му преходни вероятности от матрицата P) и не зависят от историята на процеса, довела до попадане в s_i . Повече от процесите в КС притежават тези свойства.



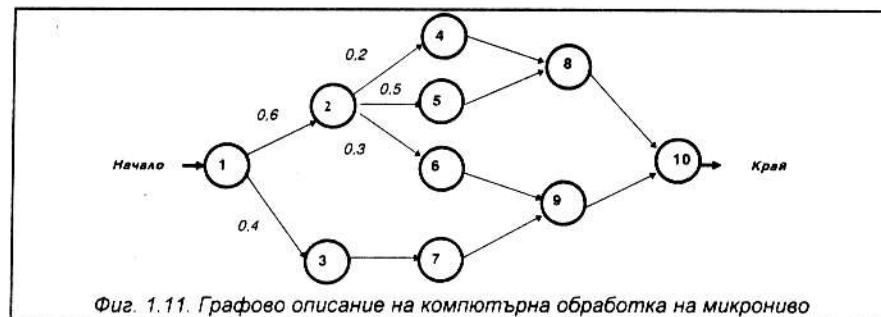
1.3.3. Организация на изчисленията

Организацията на компютърната обработка е свързана с предоставяне на изчислителни ресурси на активните приложения (задачи). Функциите по това се поемат от системните програми. При изпълнение всяка задача преминава през различни състояния в зависимост от функционалното поведение на компютъра, заложено в т.нар. глобален функционален алгоритъм. Той организира изчисленията и управлява преходите на задачите между състоянията, в които попадат. Тази концепция съответства на принципа на централизираното управление, при който всички изчислителни функции се ръководят от централен възел на базата на съхранена в паметта програма.

Организацията на изчисленията обхваща двете йерархични нива в компютърната обработка, свързани с нейното дефиниране чрез програма от високо (макро-) ниво и нейната реализация чрез машинните средства от ниско (микро-) ниво.

♦ При организация на изчислителната обработка на **микро-ниво** основните единици са елементарните операции при решаване на дадена задача. Това е нивото на машината и е скрито от потребителя. Фактически работата на компютъра по поддържане на изчисленията се осъществява на това ниво - на него са реализирани поддържащите алгоритми, управлението и пр. Ето защо се нарича още машинно ниво на работа.

Примерно описание на компютърната обработка на микро-ниво чрез насочен граф е дадено на фиг.1.11. Отделните върхове в графа представят микроинструкциите, а ребрата - разрешените преходи между тях със съответните вероятности за осъществяването им.



Всяка машинна програма (микропрограма) се състои от определен брой машинни инструкции (микроинструкции), които се изпълняват в определен ред в зависимост от логически условия. Така в микропрограма се формират няколко възможни последователности от изпълнявани микроинструкции в зависимост от допълнителни условия на изчислението.