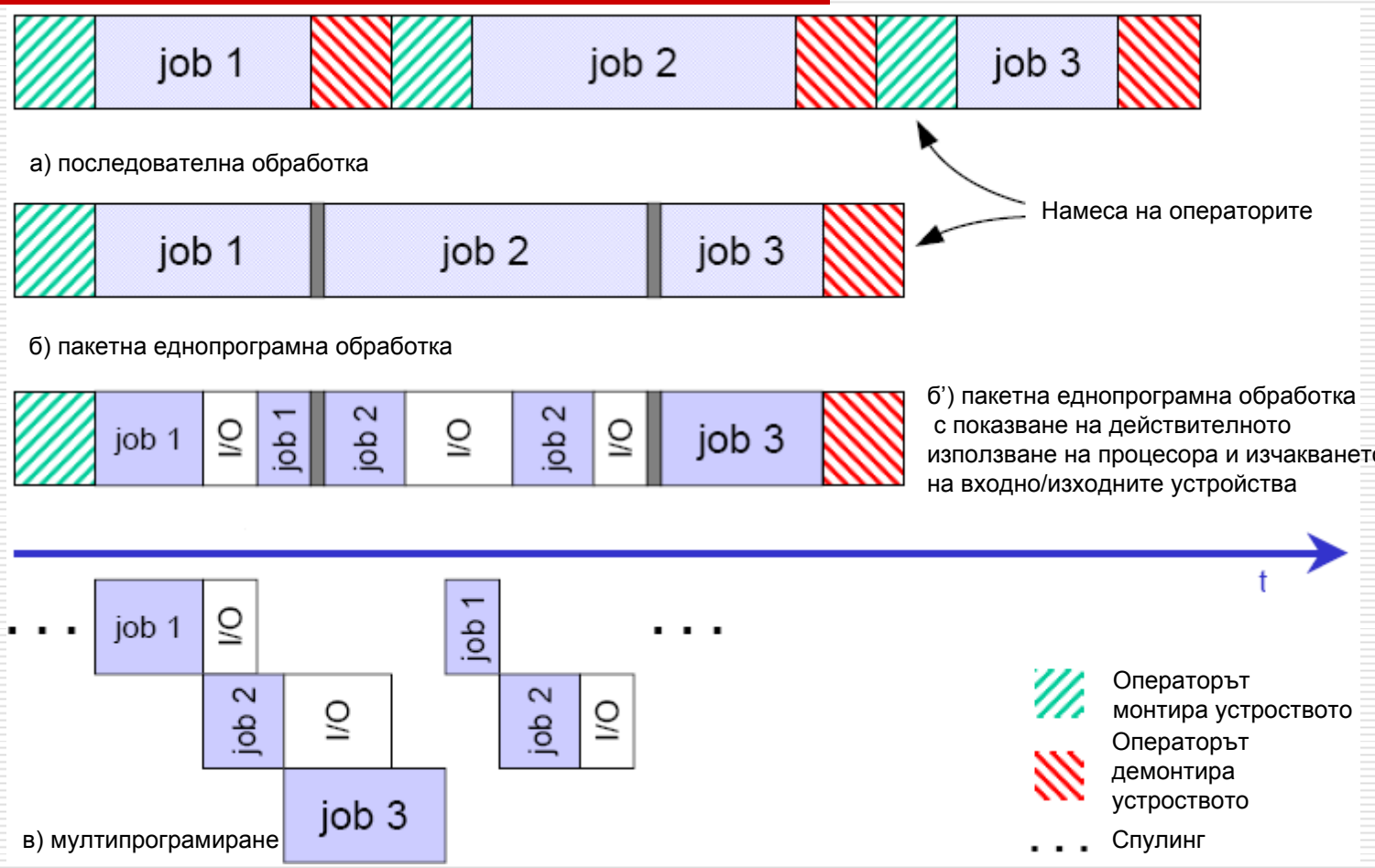
2. Процеси

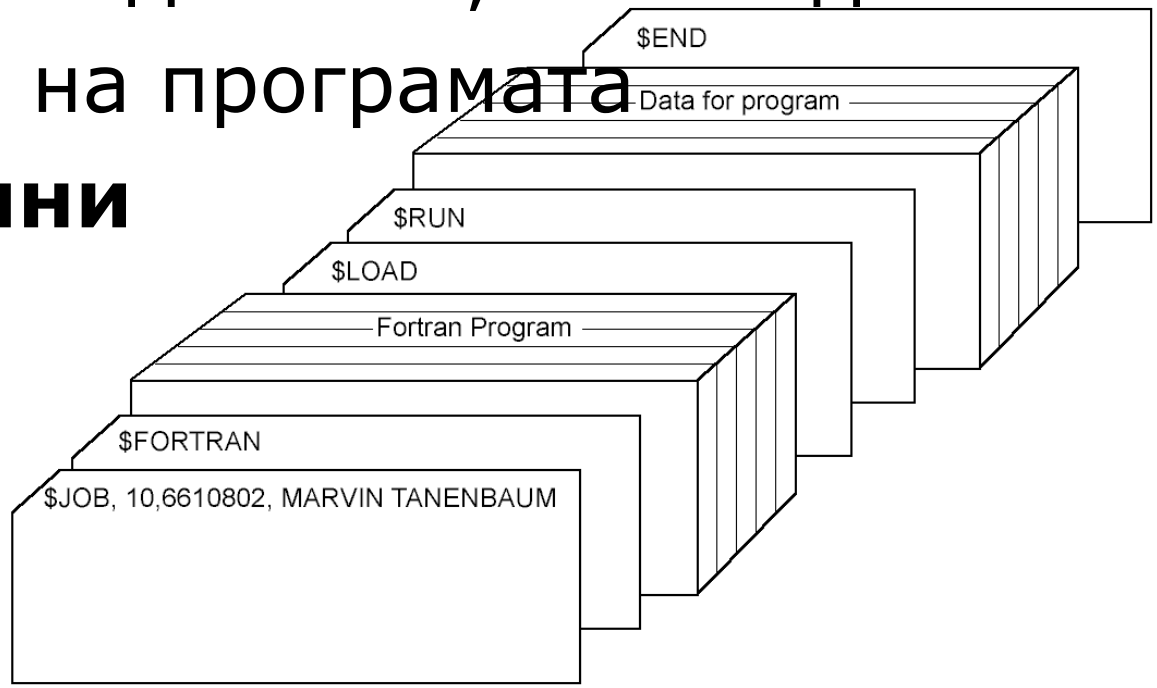
1. Понятие за процес
2. Състояния на процесите
3. Операции за работа с процесите
4. Блок за управление на процеса. Контекст на процеса
5. Еднократни операции
6. Многократни операции

Операционната система планира заданията



Понятието „задание” включва

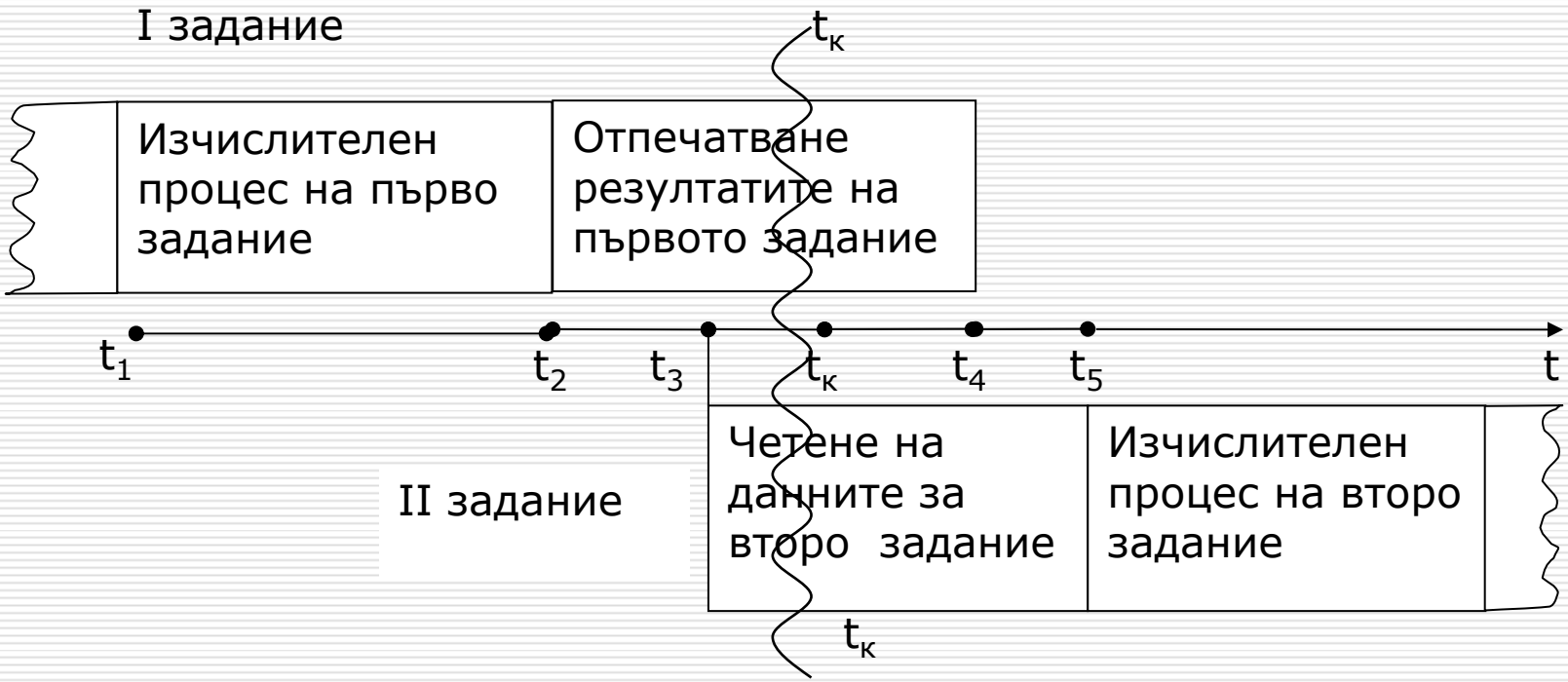
- набор от **команди** на езика за управление на заданията, необходими за изпълнение на програмата
- **входни данни** за програмата



Пакети за управление на заданията за решаване на една и съща задача с различни входни данни

Първи пакет		Втори пакет	
Управление на задание 1	**Данни на задание 1**	**Управление на задание 2**	**Данни на задание 2**

Идентични задания с еднакви входни данни, но стартирани в различен момент



Има ли идентичност на заданията във всеки от случаите?

- **Не**, т.к. двете задания се намират в различни фази на своето изпълнение
- Понятието „**задание**” ***не може*** да се използва за описание на процесите в компютъра

Синоними на „задача”

- програма в стадий на изпълнение;
- асинхронна работа;
- „живата душа” на процедурата;
- „концентрация на средствата за управление” за изпълнение на процедура;
- нещо, представено във вид на „блок за управление на процеса” на операционната система;
- обект, който се заделя за процесора;
- „диспечериран” модул

Понятието „процес”

- За пръв път той се въвежда от разработчиците на системата MULTICS през 60-те години
- **процесът е абстрактно понятие, описващо работата на програмите**

Какво е процес?

•Процесът е дейност по изпълнение на програмата

Програма

Салата със спагети

Продукти:

100 г сварени спагети
200 г коктейлни домати
1 краставица
100 г черни маслини без костилки

връзка босилек
1-2 с л винен оцет
2-3 с л зехтин
1 скилидка чесън
сол
смлян черен пипер

Нишка на изпълнението

Приготвяне:

• Доматите се нарязват на четвъртинки, а краставицата на дребни кубчета.
• Маслините и босилекът се накълцват на дребно.
• В голяма купа се смесват подготвените продукти.
• Заливат се със заливка и се разбъркват.

Процес

Процесор

Продукти

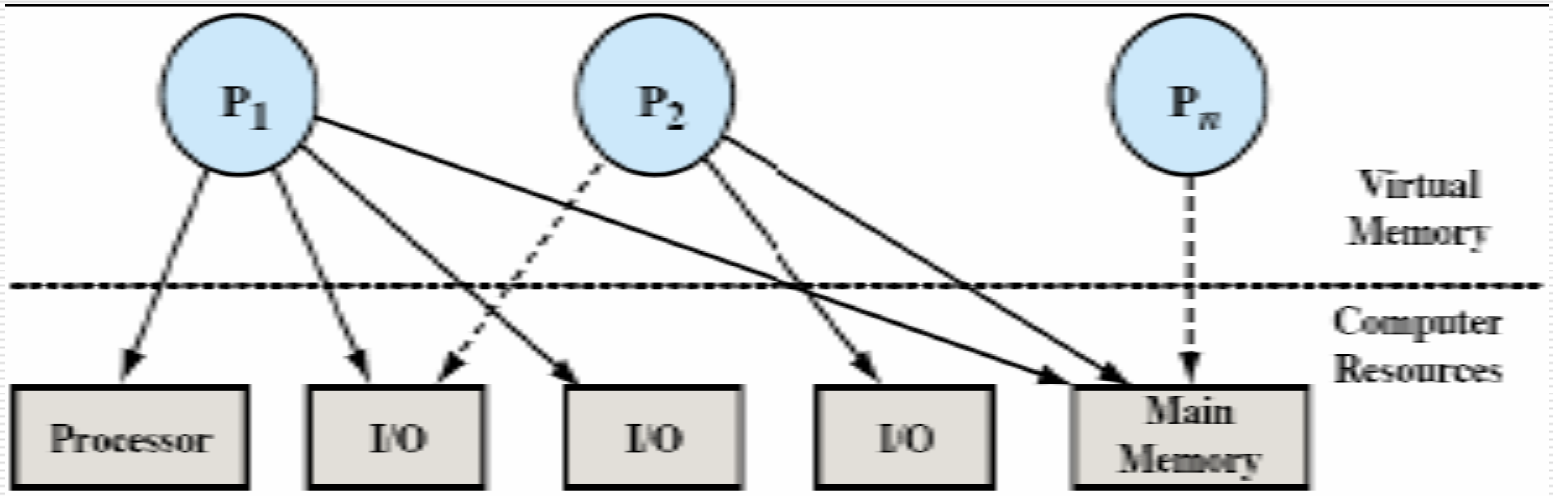
Определение

- **Процесът представлява** съвкупност от **команди**, асоциирани с тях **ресурси** в текущия момент на неговото изпълнение, намиращи се **под управлението на операционна система**.

В мултипрограмната ОС се създават много процеси

Всеки процес, в хода на неговото изпълнение се нуждае от достъп до системните ресурси:

процесор, оперативна памет, I/O устройства

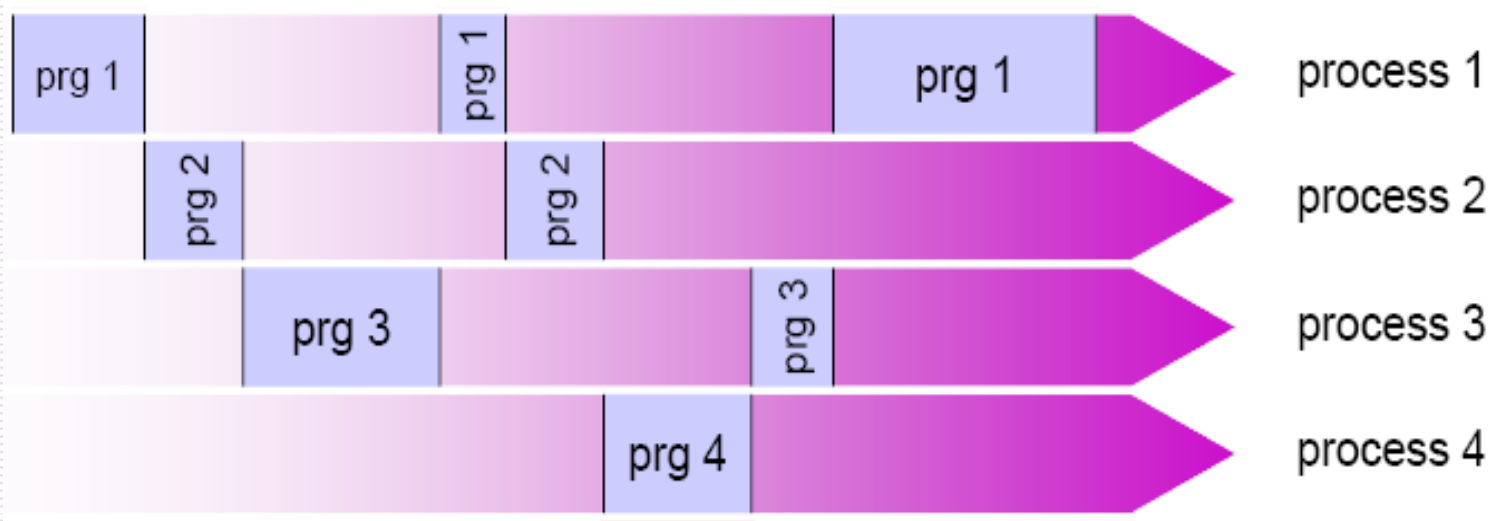


Разпределението на ресурсите при процесите (една моментна снимка)

Многозадачността може да представи като многобройни процеси стартирани (псевдо) паралелно

Многозадачност от гледна точка на процесора

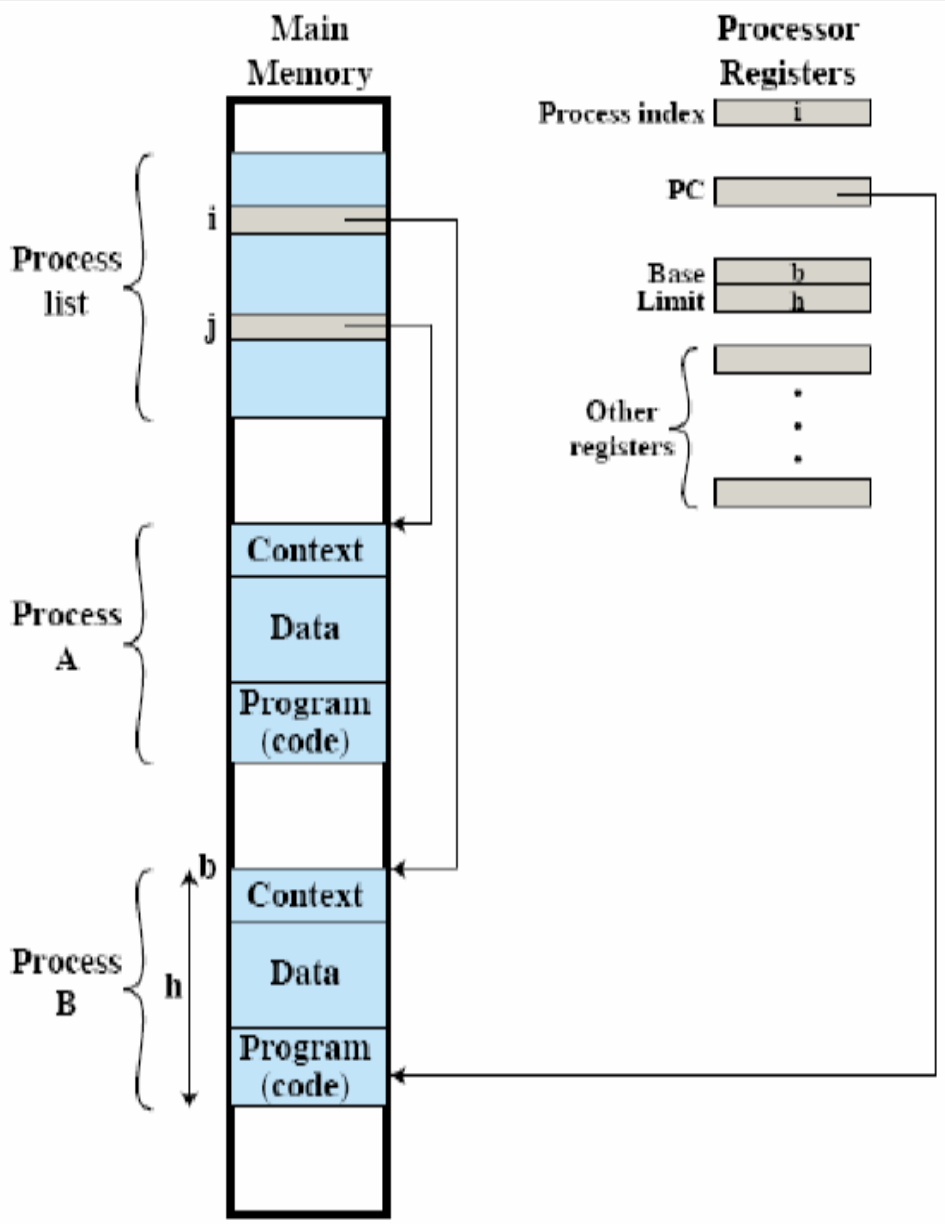
Многозадачност на процесите от гледна точка на 4 програми



Процесът се състои от три компонента

Потребителско адресно пространство

1. Изпълнима програма
2. Асоциирани данни, необходими на програмата
3. **Контекст** на изпълнението на процеса, който съдържа цялата информация от която има нужда ОС за да управлява процеса (ID, състояние на процеса, CPU регистри, стек, и т.н.)

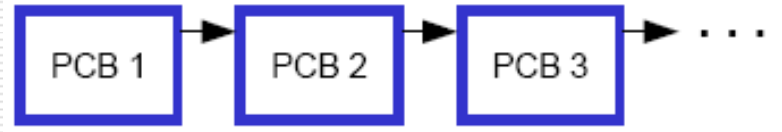


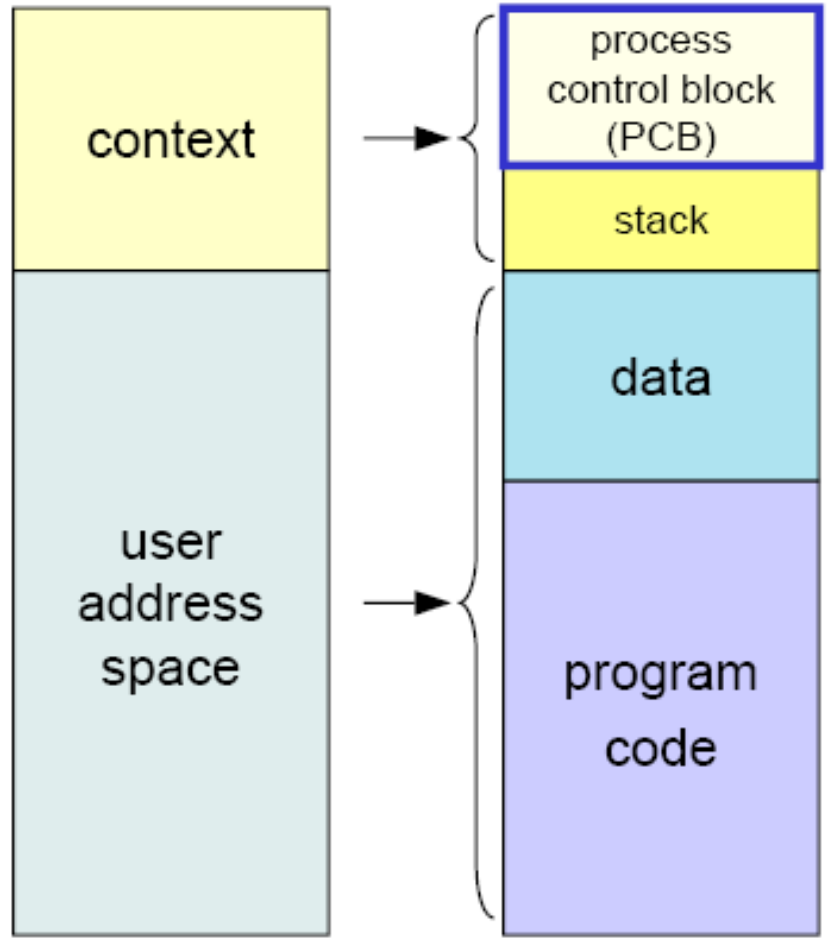
Блок за управление на процеса Process Control Block (PCB)

- състояние на процеса;
- програмен брояч на процеса (това е адресът на следващата за изпълнение команда);
- съдържание на регистрите на процеса;
- данни, необходими за планиране използването на процесора и управлението на паметта (приоритет на процеса, размер и разположение на адресното пространство и т.н.);
- отчетни данни (идентификационен номер на процеса, кой потребител го е създал, общо време за използване на процесора от дадения процес и т.н.);
- сведения за входно-изходните устройства, свързани с процеса (например, кои устройства са отдадени за работа с процеса, таблица на отворените файлове).

Блокът за управление на процеса

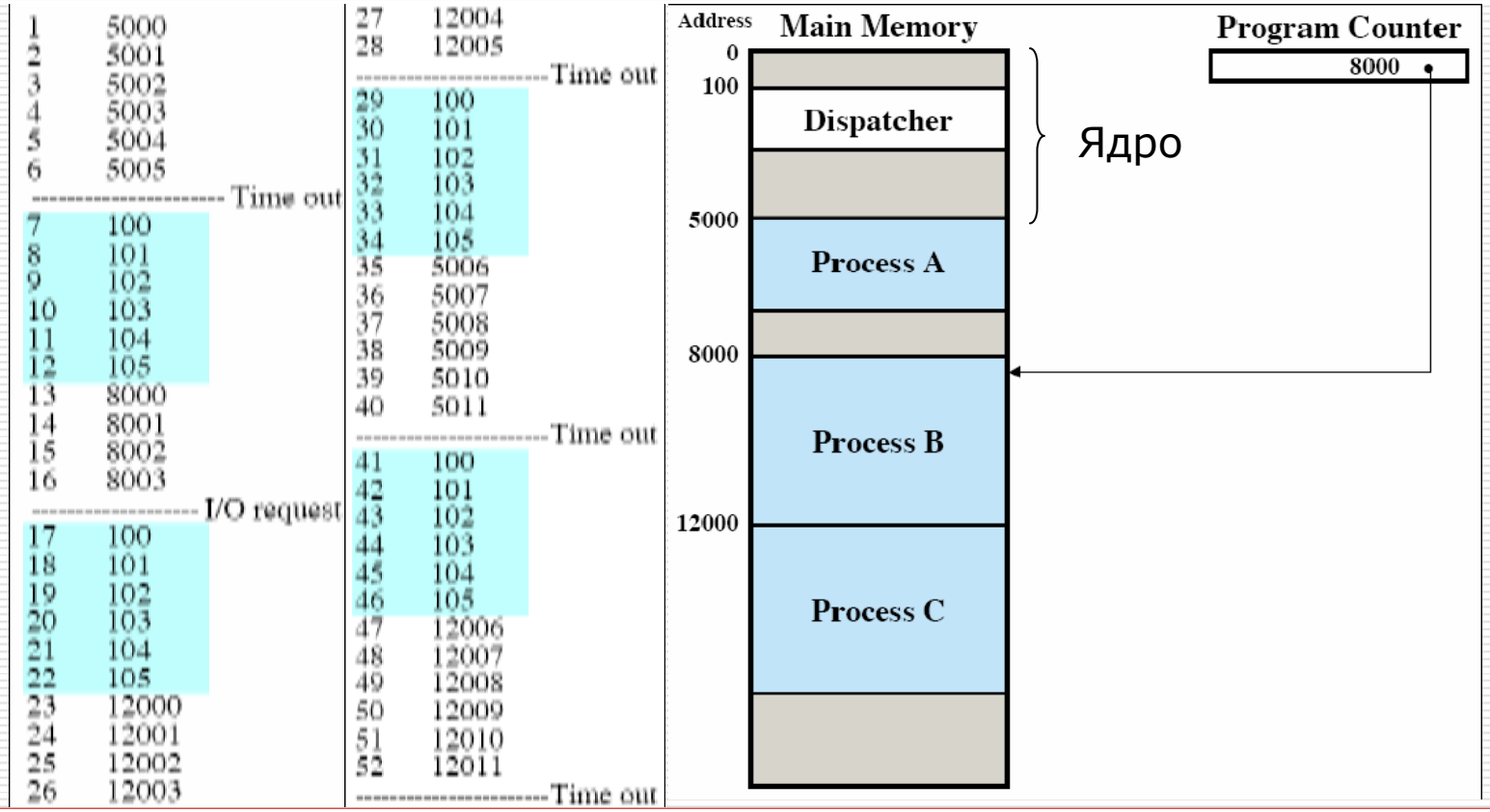
- е включен в контекста, заедно със стека;
- е "моментна снимка", която съдържа всички необходими и достатъчни данни за да се рестартира процеса когато е необходимо;
- е една входна точка в таблицата на процеса (масив или свързан списък)





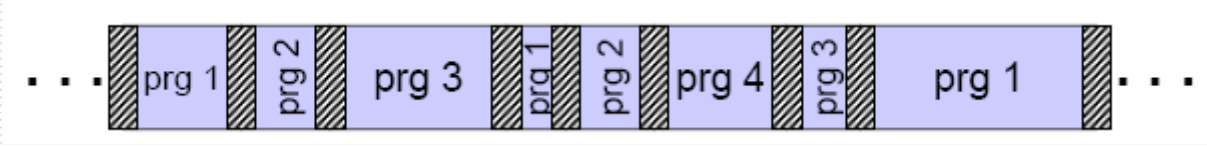
Диспечерът превключва процесора между процесите

Диспечерът е обикновена програма в ядрото на ОС

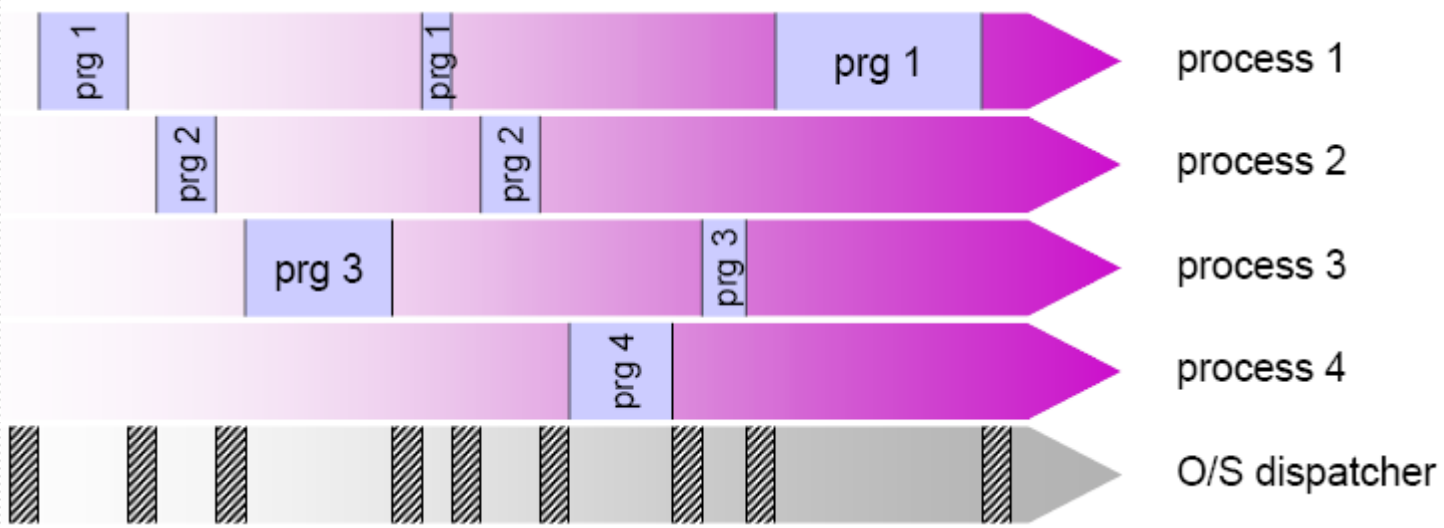


Диспечерът превключва процесора между процесите

Многозадачност от гледна точка на процесора

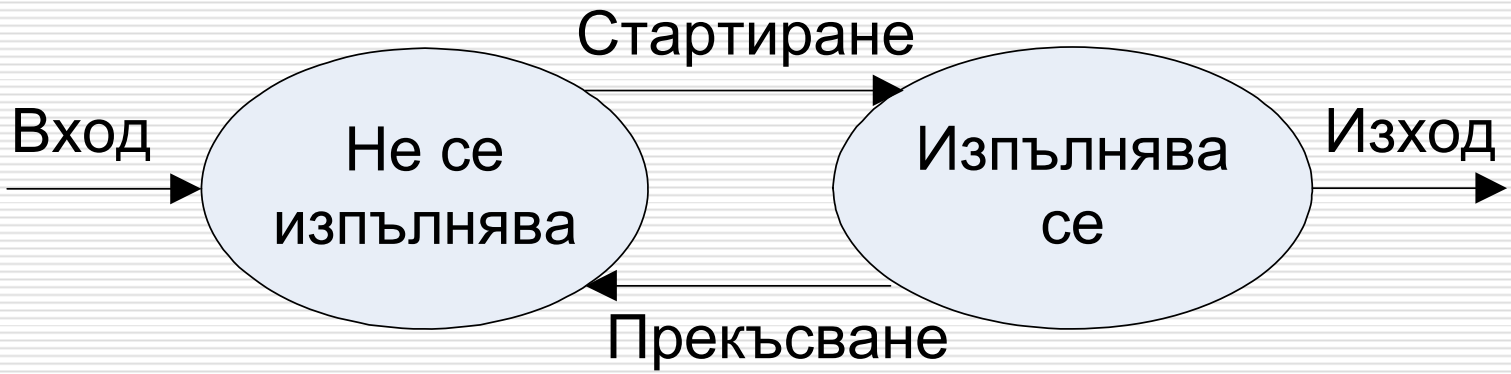


Многозадачност на процесите от гледна точка на 4 програми



Състояния на процесите

По всяко време, всеки процес или се изпълнява от процесора или не се изпълнява



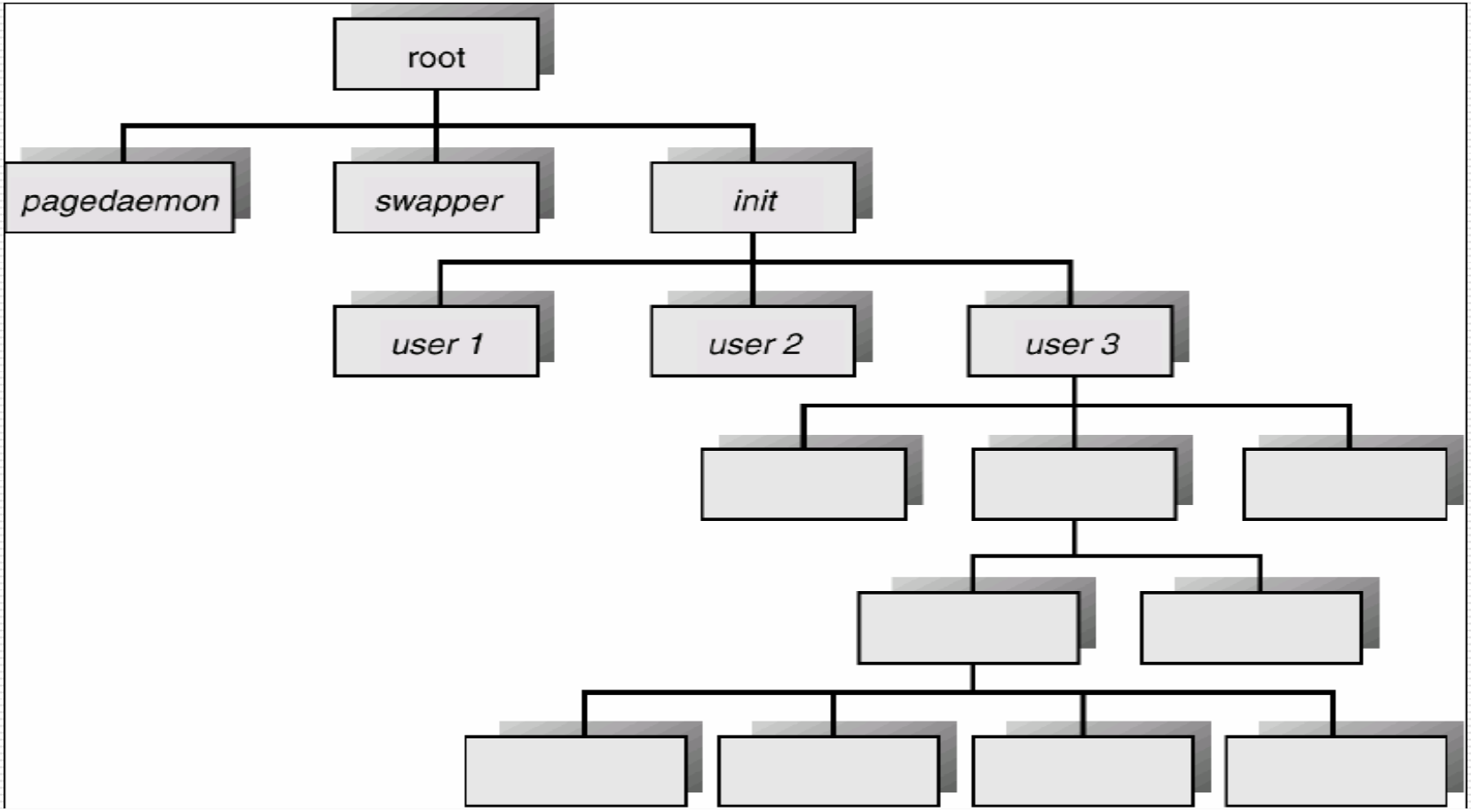
Някои събития които водят до създаване на процес (1)

- **При зареждане на системата**
  - Когато системата е инициализирана, се стартират различните фонови процеси или "демони" (email, logon, и т.н.)
- **Потребителят извиква приложение за изпълнение**
  - потребителят може да създаде нов процес като пише команда в шела или щракне с мишката върху приложение

Някои събития които водят до създаване на процес (2)

- **Съществуващият процес създава процес-наследник**
  - Например, процесът на сървъра (принтер, файл) може да създаде нов процес за всяка заявка, която той обработва
  - ***Init*** демонът очаква потребителския login и създва шел
- **Системата за пакетна обработка поема за обработка следващата задача в редицата**

Дърво от процеси в типична UNIX система



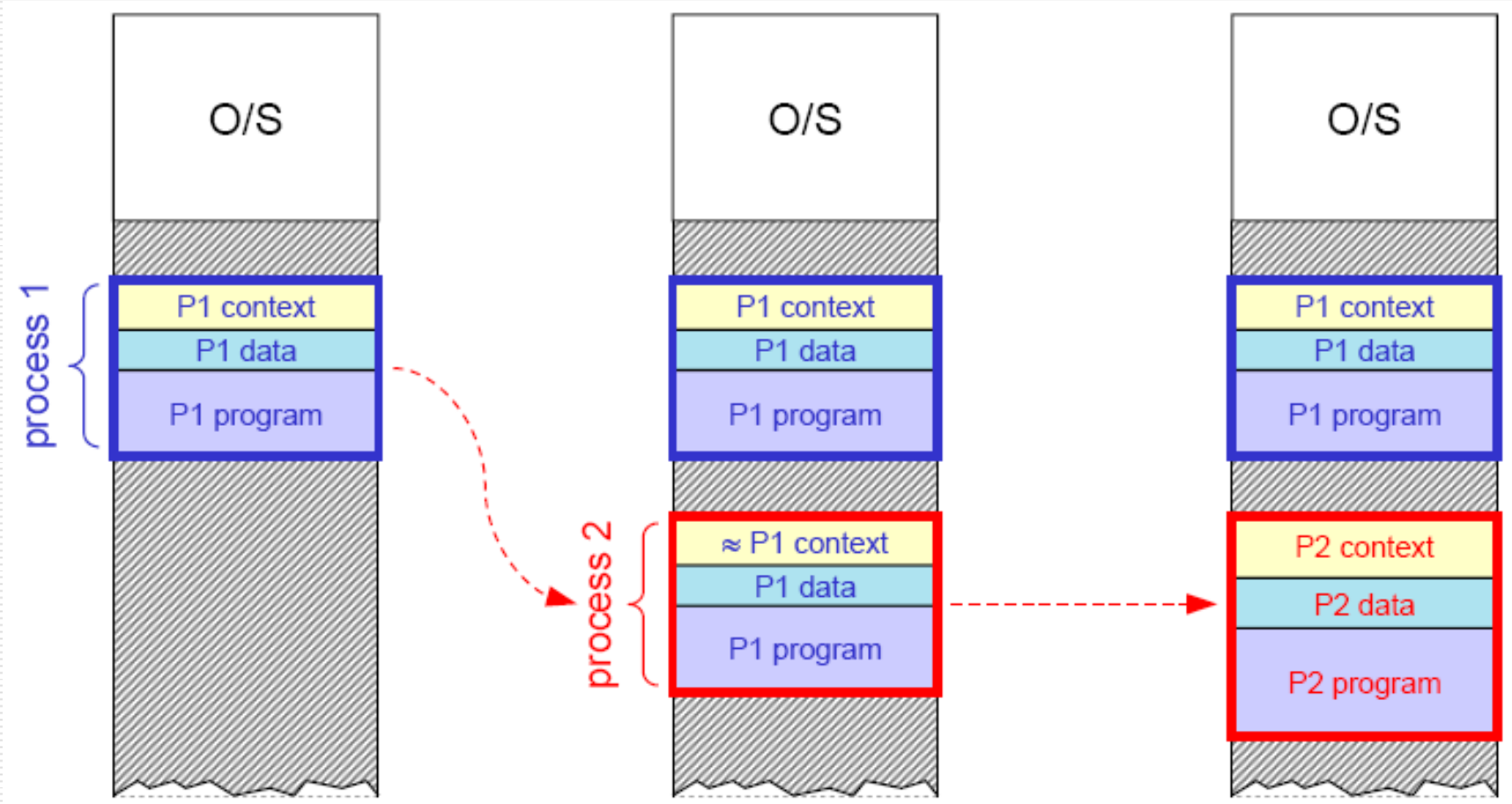
```
int main(...)
{
   ...
   if ((pid = fork()) == 0)   // Създаване на процес
   {        fprintf(stdout, "Child pid: %i\n", getpid());
            // Прекъсване на процеса-наследник
            err = execvp(command, arguments);
            fprintf(stderr, "Child error: %i\n", errno);
            exit(err);
   }
   else if (pid > 0)
  {
            // Тук сме в родителския процес
            fprintf(stdout, "Parent pid: %i\n", getpid());
            pid2 = waitpid(pid, &status, 0);
            // Изчакваме процеса-наследник
            ...
   }
   ...
   return 0;
}
```

Клониране на процес-наследник

pid = fork()

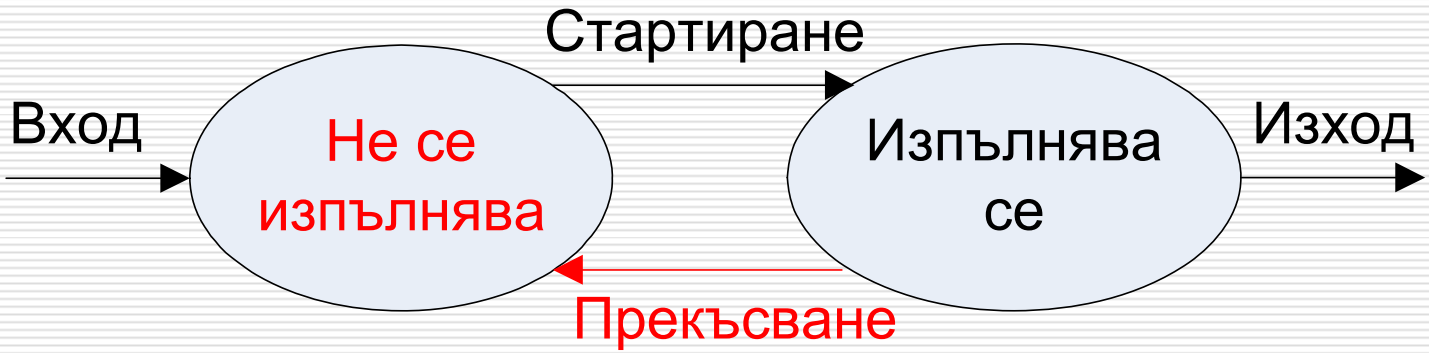
Изпълненние на процеса-наследник

execve (name, ...)



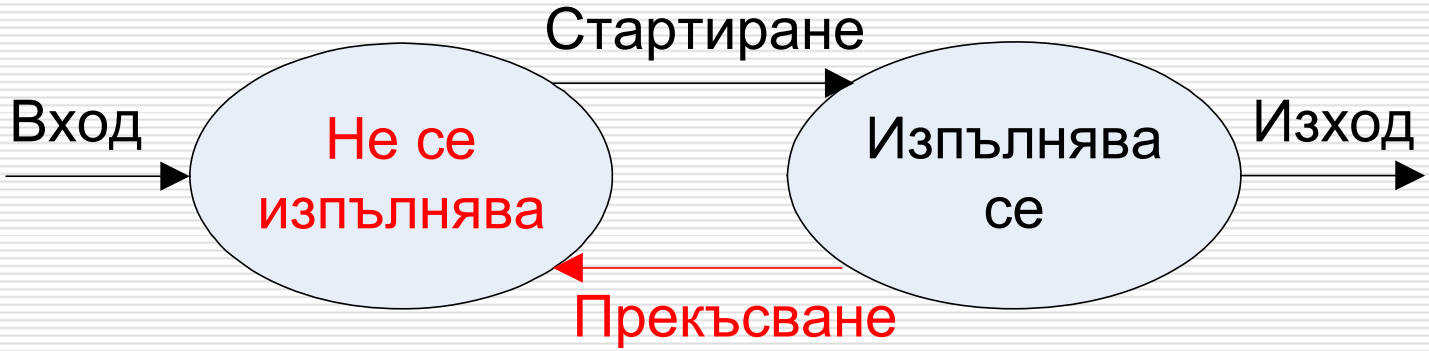
Някои събития, които прекъсват изпълнението на процеса (1)

- **I/O изчакване**
  - Процесът се обръща към I/O системно извикване, което блокира обръщенията към I/O устройство. ОС установява процеса в режим "**Не се изпълнява**" и избира друг процес за изпълнение.



Някои събития, които прекъсват изпълнението на процеса (2)

- **Приоритетно прекъсване**
  - Изпълнението се спира и процесът се връща в режим "**Не се изпълнява**" поради изтичане на кванта време или поради появата на по-приоритетен процес.



Някои събития, които водят до завършване на процеса (1)

- Нормално завършване със или без код за грешка
  - Процесът доброволно изпълнява системното извикване **exit (err)** и съобщава на ОС, че е завършил

Някои събития, които водят до завършване на процеса (2)

- Фатална грешка
  - сервизни грешки: няма налична памет за разпределение, I/O грешка, и т.н.
  - лимитът на общото време е изчерпан
  - аритметична грешка, обръщение извън-границите на паметта, и т.н.

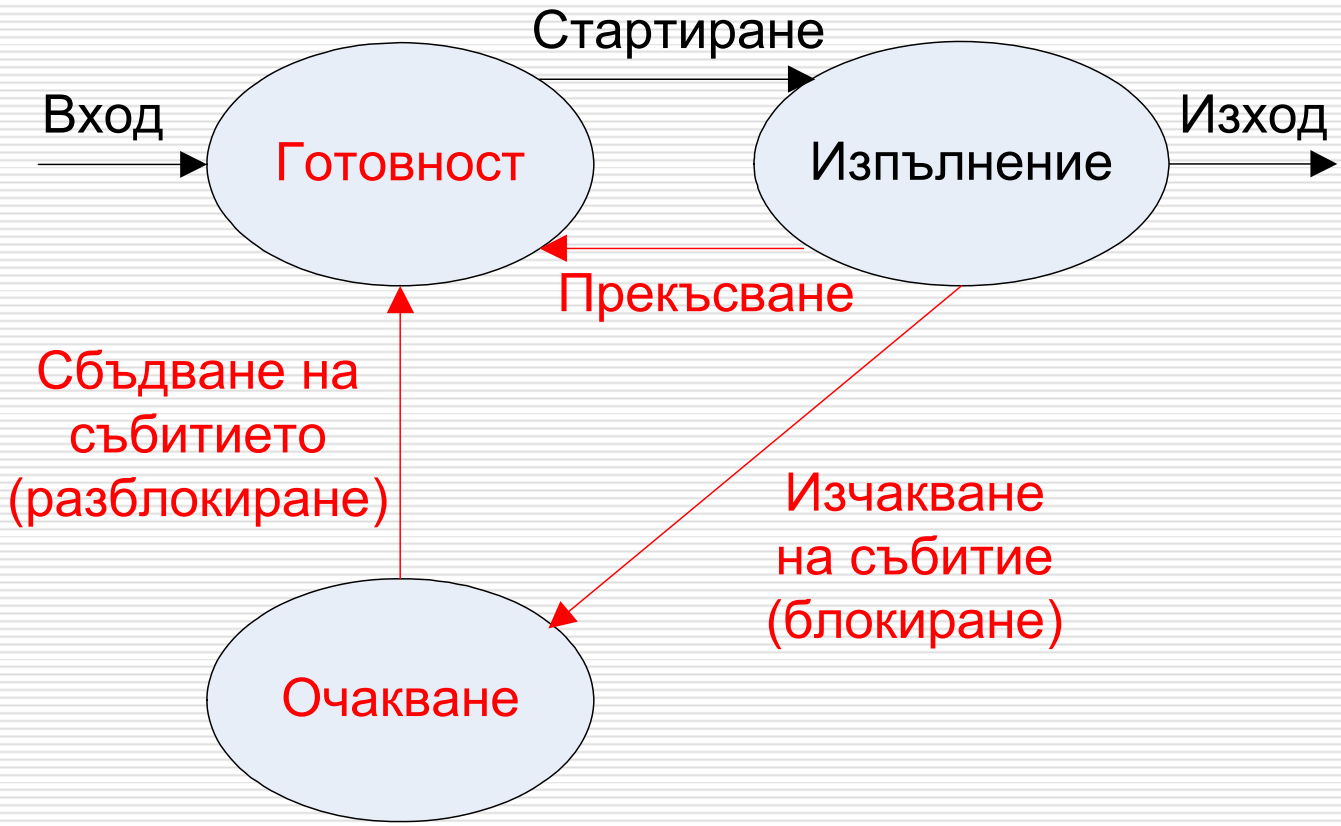
Някои събития, които водят до завършване на процеса (3)

- Унищожен от друг процес чрез ядрото
  - процесът получава сигнал SIGKILL
  - в някои системи, ако родителски процес бъде унищожен, също така се унищожават и всички негови наследени (подчинени) процеси

Проблем с модела от две състояния

- Някои "**неизпълнявани**" процеси са блокирани (в очакване на I/O операция, и т.н.)
- ОС губи време за сканиране на опашката от готови процеси

Решение: да се раздели “неизпълняваният” на “готов” и “блокиран”



Модел на преходите на процесите с 3 състояния

Някои събития които водят към блокиране на процесите

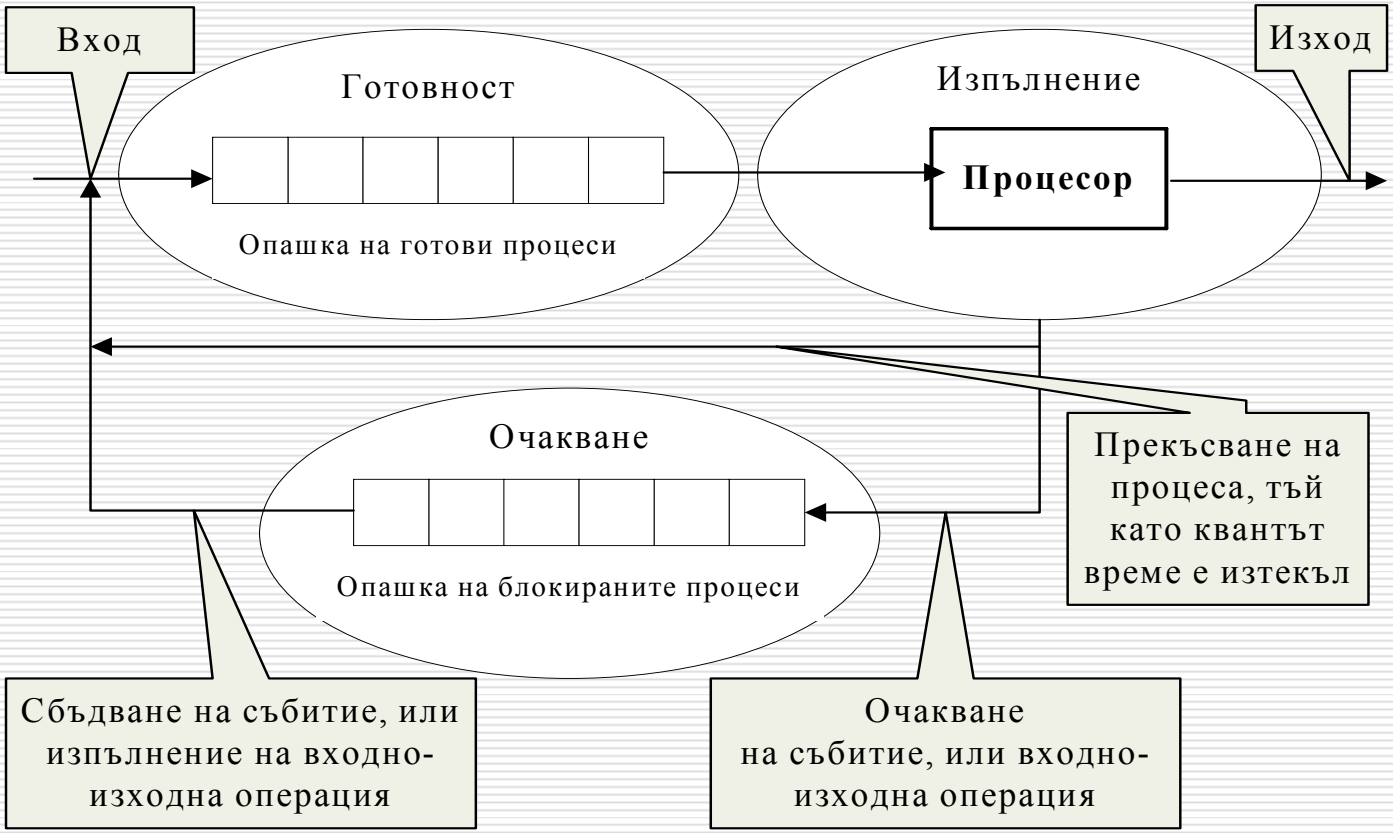
- **I/O изчакване**
  - Процесът използва I/O системно извикване, което блокира чакащите за I/O устройство: ОС установява процеса в състояние "**блокиран**" и стартира следващия процес

Някои събития които прекъсват процесите

- Процесът се прекъсва поради изтичане на времето и се връща в състояние "**готовност**". ОС предава друг процес към процесора.

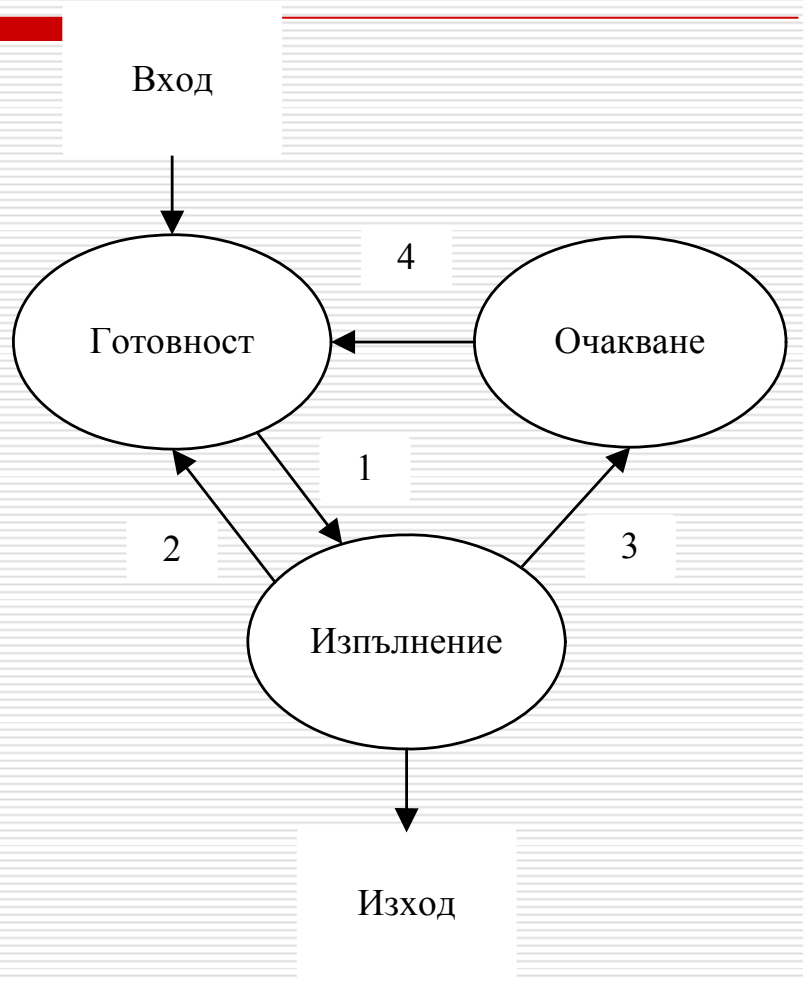
Как ОС следи трите състояния на процеса?

Като поддържа допълнителна опашка на блокираните процеси



Подробна диаграма на състоянията на процеса

1. Процесът е избран за изпълнение
2. Прекъсване на процеса
3. Очакване на събитие, или входно-изходна операция
4. Настъпило е събитие, или чаканата входно-изходна операция е приключила



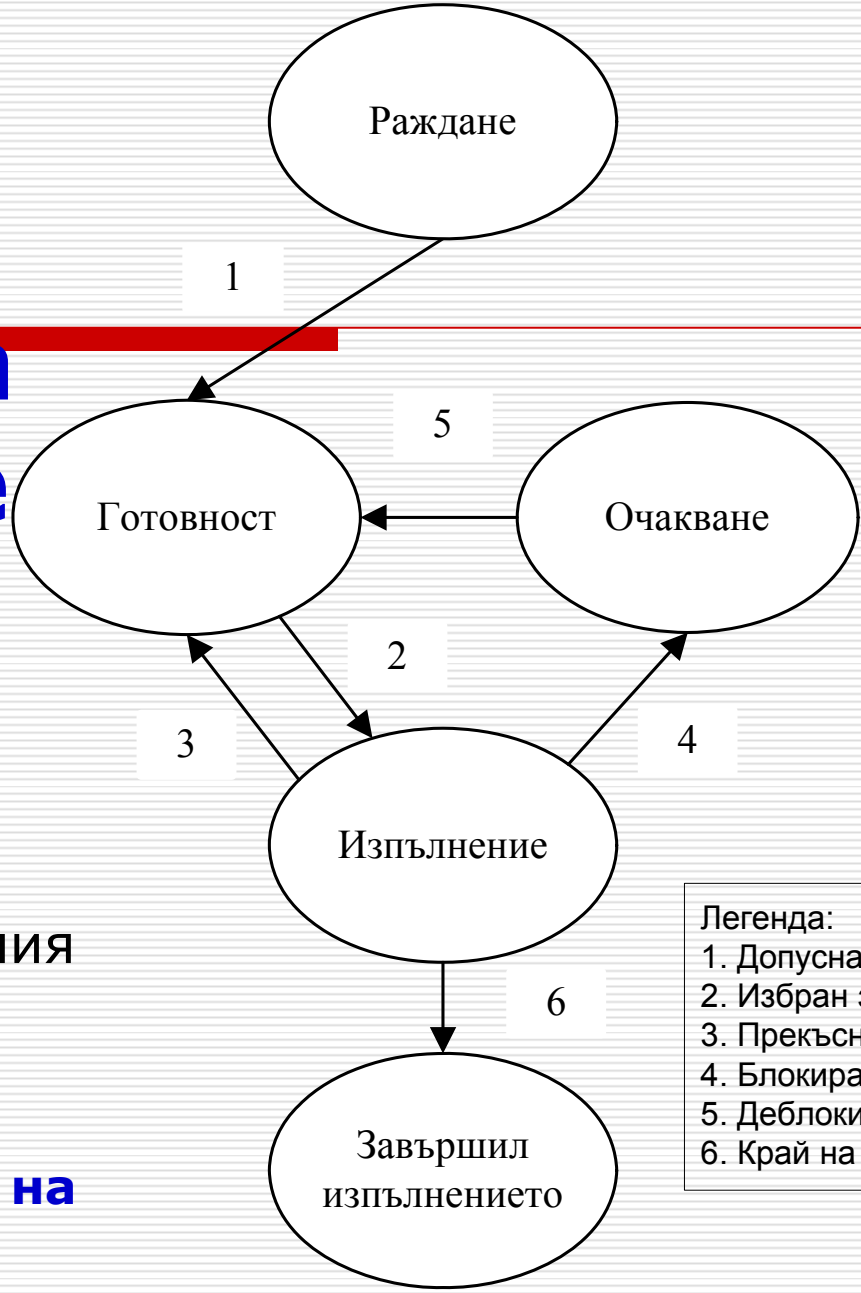
Прекратяване на процес
(по 2 причини)

- за неговата по-нататъшна работа трябва да **чака сбъдване на събитие** (например завършване на входно-изходна операция);
- **изтекъл е квантът време**, определен от операционната система за работата на дадения процес.

Излизане от състояние "изпълнение"

- процесът е **завършен** и операционната система го прекъсва;
- процесът **не може да продължи**, докато не се сбъдне някакво събитие и операционната система го привежда в състояние „очакване”;
- в резултат на възникване на **прекъсване** в компютъра (например изтекъл е квантът време, отделен за дадения процес) и операционната система го връща в състояние „готовност”.

Подробна схема на състоянията и преходите на процеса



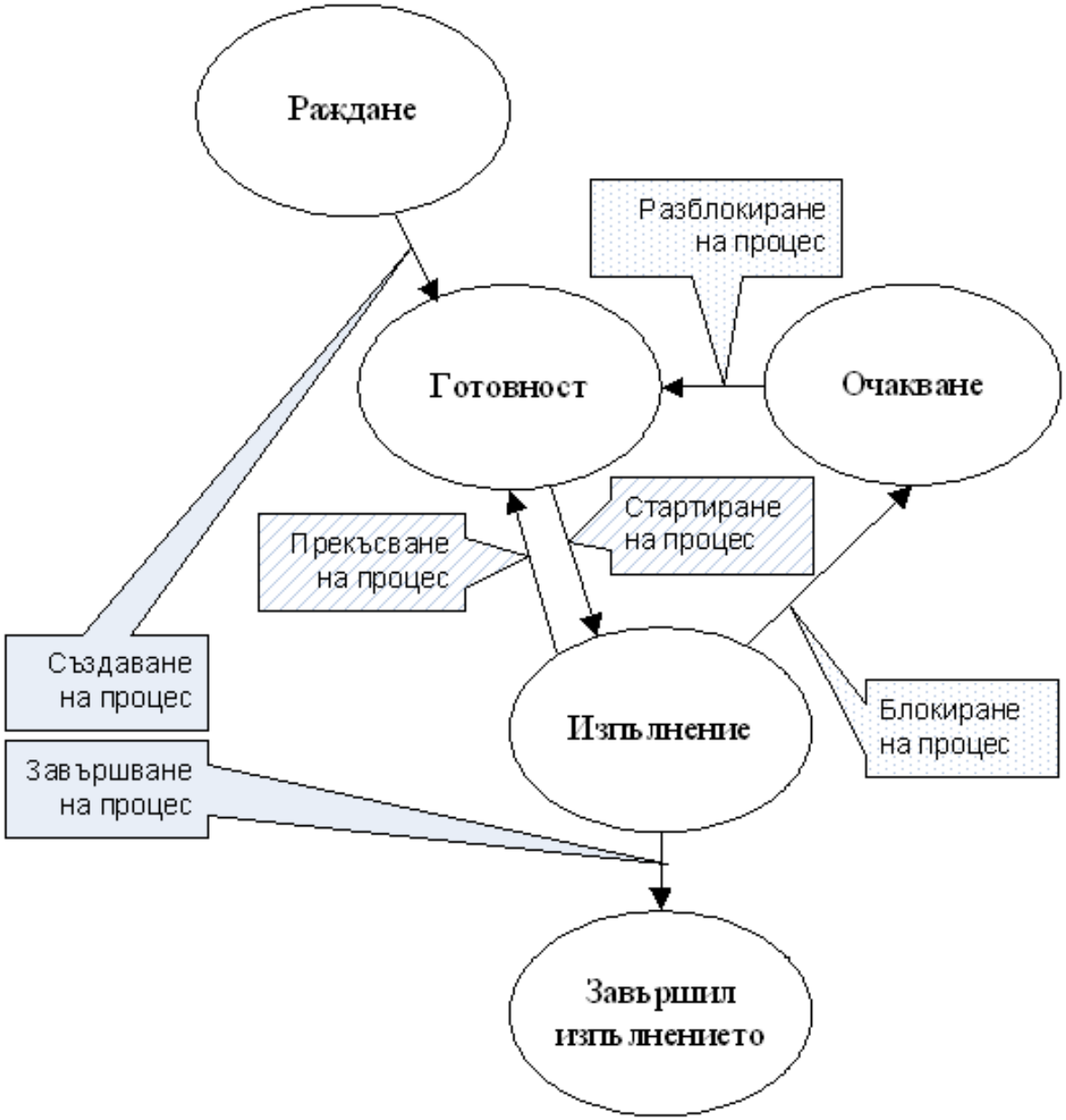
Легенда:
1. Допуснат за планиране
2. Избран за изпълнение
3. Прекъснат
4. Блокиран
5. Деблокиран
6. Край на изпълнението

Windows NT - 7 състояния

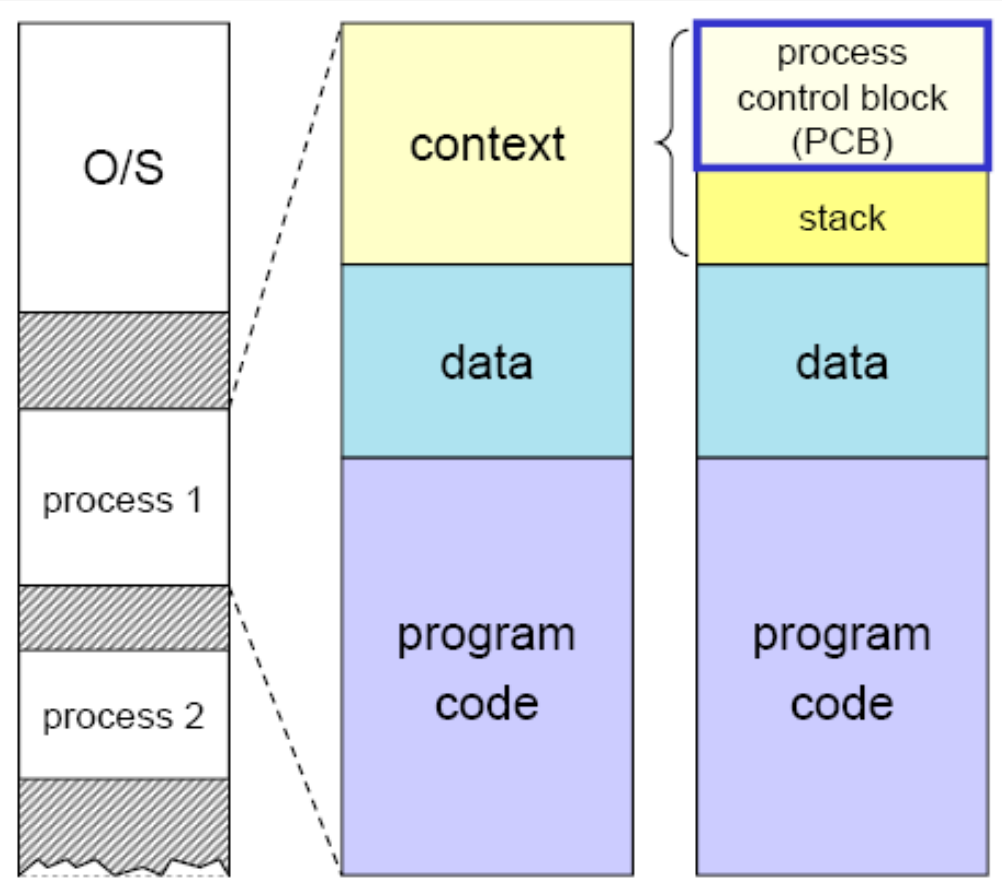
UNIX – 9 състояния

Всички операционни системи се подчиняват на изложения модел.

Преходите се обединят в три противопо-ложни по характер двойки



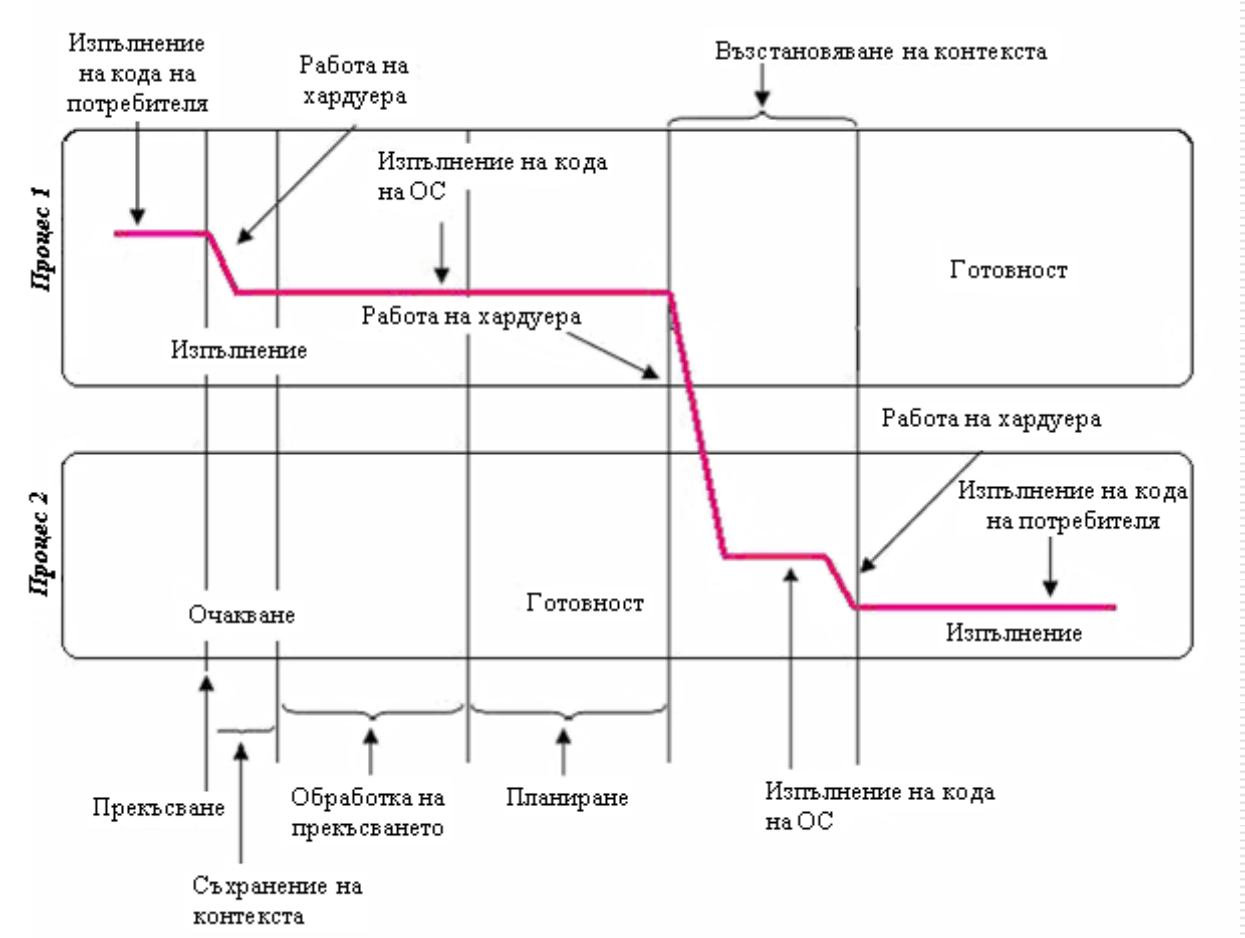
Пример за процес и PCB - разположение в паметта



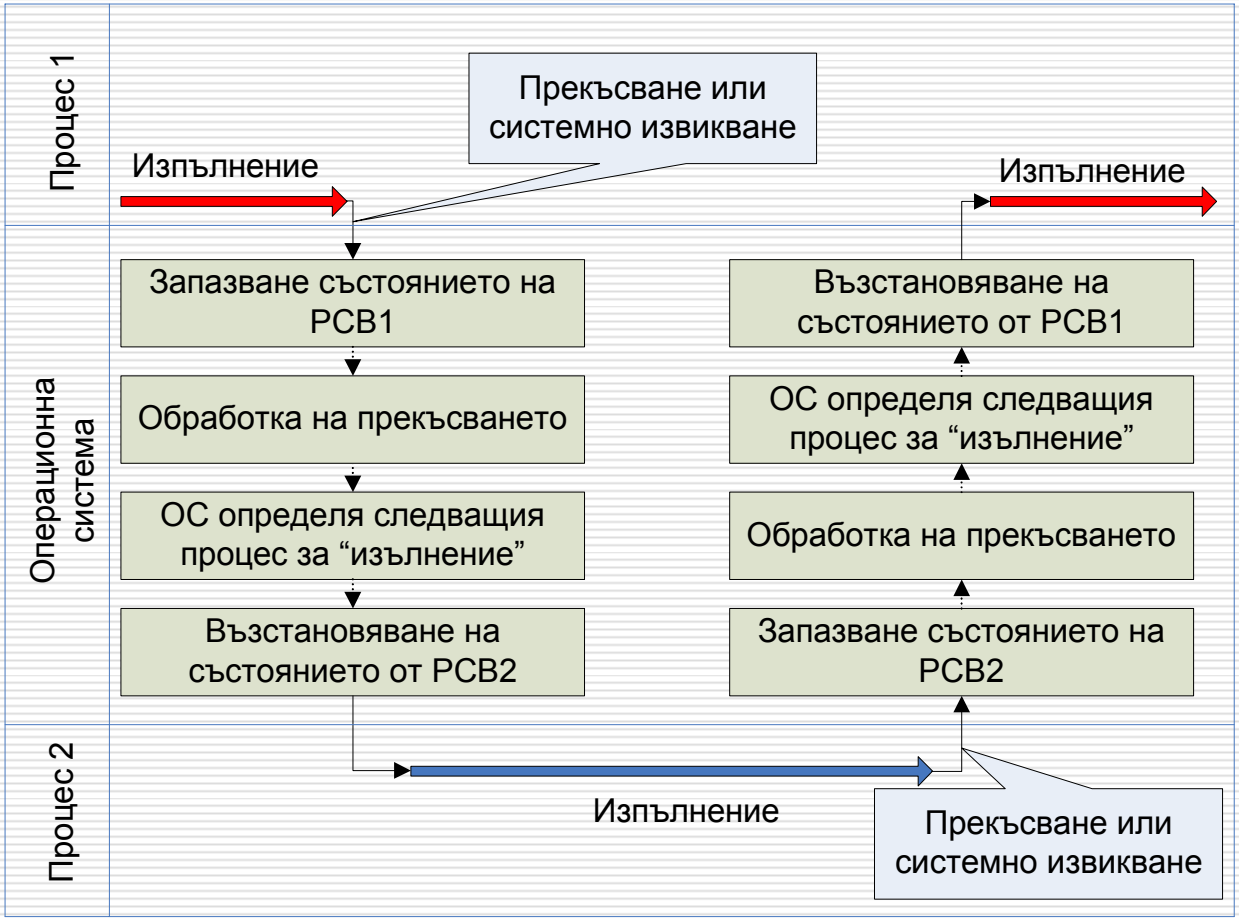
PCB включва 3 секции:

- **1. Идентификация**
  - ID на процеса
  - Родителски идентификатор
  - Потребителски идентификатори, и т.н.
- **2. Състояние на процесора**
  - Регистри видими за потребителското приложение
  - Управление и регистри на състоянието
  - Стек с указатели, и т.н.
- **3. Управление**
  - Информация за планирането и състоянията
  - Връзки към други процеси
  - Привилегии на паметта, и т.н.

Превключване на контекста



Превключване на контекста



Обобщение (1)

- **Процес** е съвкупност от:
  - **команди** за изпълнение;
  - асоциирани **ресурси** за времето на изпълнение, намиращи се **под управлението на операционната система**.

Обобщение (2)

- Процесът напълно **се описва** със свой **контекст**, състоящ се от:
  - регистрова част;
  - системна част;
  - потребителска част.

Обобщение (3)

- Процесите могат да се намират в **пет основни състояния**:
  - раждане;
  - готовност;
  - изпълнение;
  - очакване;
  - завършил изпълнението.

Обобщение (4)

- ОС привежда процесите от едно състояние в друго, като извършва над тях определени операции:
  - създаване на процес;
  - завършване на процес;
  - прекъсване на процес;
  - стартиране на процес;
  - блокиране на процес;
  - разблокиране на процес;
  - изменение на приоритета на процеса.

Обобщение (5)

- Превключването на контекста няма отношение към полезната работа, изпълнявана от процесите, и времето, изгубено за него, съкращава полезното време за работа на процесора.

Следва

продължение