

# Процеси: същност, състояния, управление — 116

---

## 1. Какво е процес

Процесът е програма в изпълнение. Когато стартираш програма, тя става процес. Процесът е "живата" версия на програмата - програмата е файл на диска, а процесът е тази програма докато работи.

### Разлика между програма и процес:

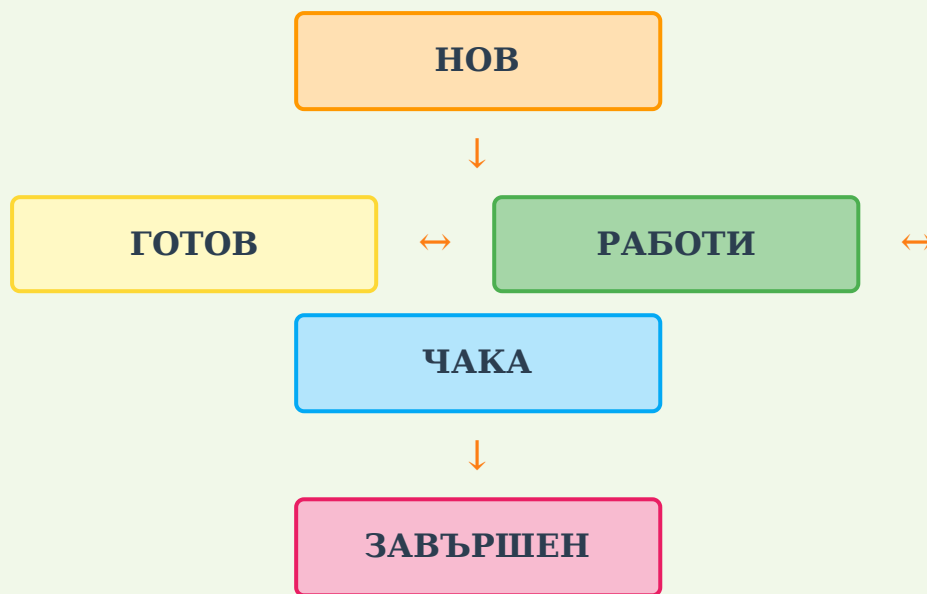
- **Програма** - файл на диска, не прави нищо докато не я стартираш
- **Процес** - програмата докато работи, има памет, използва процесор

### Прости примери:

- **Firefox на диска** - програма (файл)
- **Firefox отворен** - процес (работи)
- **Word на диска** - програма
- **Word отворен с документ** - процес
- **Игра инсталирана** - програма
- **Игра която играеш** - процес

## 2. Състояния на процеса

### Петте състояния на процеса:



#### 1. НОВ (New)

- Процесът току-що е създаден
- Все още не е зареден напълно
- ОС го подготвя за работа

#### 2. ГОТОВ (Ready)

- Готов е да работи
- Чака процесорно време
- Стои на опашка

#### 3. РАБОТИ (Running)

- В момента използва процесора
- Изпълнява инструкции
- Само един процес на процесор

#### 4. ЧАКА (Waiting)

- Чака нещо да се случи
- Например да се отвори файл
- Или данни от мрежата

#### 5. ЗАВЪРШЕН (Terminated)

- Процесът е приключил
- ОС го изчиства
- Освобождава ресурсите

### 3. PCB - Process Control Block

#### Какво е PCB:

PCB е като личната карта на процеса. Операционната система пази информация за всеки процес в специална структура - Process Control Block.

### Какво съдържа PCB:

**Process ID (PID) - уникален номер**

**Състояние - Готов, Работи, Чака...**

**Program Counter - къде сме стигнали**

**Регистри на CPU - текущи стойности**

**Памет - колко и къде**

**Отворени файлове - списък**

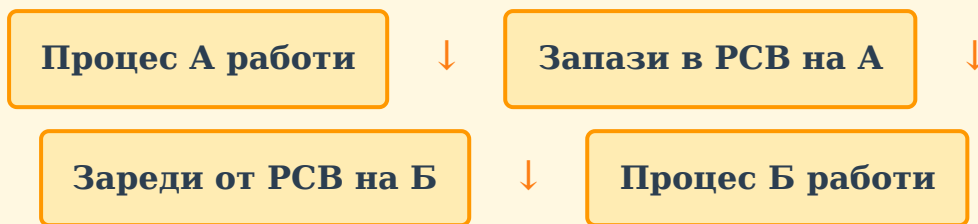
**Приоритет - колко е важен**

### Защо е нужен PCB:

- **Когато процесът спре** - ОС запазва всичко в PCB
- **Когато процесът продължи** - ОС зарежда всичко от PCB
- **Управление** - ОС знае какви ресурси използва процесът

## 4. Context Switch - превключване на контекста

### Как работи превключването:



### Стъпките на Context Switch:

- **Стъпка 1** - Спри текущия процес
- **Стъпка 2** - Запази състоянието му в PCB
- **Стъпка 3** - Избери нов процес от опашката
- **Стъпка 4** - Зареди състоянието на новия процес от PCB
- **Стъпка 5** - Стартирай новия процес

Context Switch отнема време! По време на превключването процесорът не върши полезна работа. Затова ОС се опитва да го прави не твърде често.

## 5. Многозадачност (Multitasking)

### Как компютърът "прави много неща наведнъж":

- **Реално** - процесорът работи само с един процес в даден момент
- **На практика** - превключва много бързо между процесите
- **Резултат** - изглежда че всички работят едновременно

### Примери за многозадачност:

- **Слушаш музика** докато пишеш в Word
- **Теглиш файл** докато сърфираш в интернет
- **Копираш файлове** докато гледаш видео
- **Антивирусът сканира** докато играеш игра

## 6. Планиране на процесите (Scheduling)

### Какво е планиране:

Планирането е решаването кой процес да работи следващ. Има специален компонент на ОС - scheduler (планировчик), който взема това решение.

### FCFS - First Come First Served

- Кой дойде първи, работи първи
- Най-простият алгоритъм
- Понякога неефективен

### Round Robin

- Всеки процес работи малко време
- После отива накрая на опашката
- Справедливо за всички

### Priority Scheduling

- По-важните процеси първи
- Всеки има приоритет
- Добро за системни задачи

## 7. Създаване и унищожаване на процеси

### Създаване на процес:



### Кога се създават процеси:

- **Стартиране на програма** - потребителят кликва на икона
- **От друг процес** - един процес може да създаде друг (fork)
- **Системна задача** - ОС стартира автоматично услуги
- **Планирани задачи** - cron jobs, scheduled tasks

### Унищожаване на процес:



### Причини за унищожаване:

- **Нормално завършване** - програмата си свърши работата
- **Грешка** - програмата се срина
- **Потребителят спря** - натиснал "Затвори" или Ctrl+C
- **Kill команда** - администраторът го спря насила
- **Системна грешка** - ОС спря процеса

## 8. Родителски и дъщерни процеси

### Йерархия на процесите:

- **Родителски процес** - създава нови процеси
- **Дъщерен процес** - създаден от друг процес
- **Дървовидна структура** - процесите образуват дърво

### Примери:

- **Terminal** → стартира **bash** → стартира **ls**
- **Browser** → създава **tab процеси**
- **init/systemd** → родител на всички процеси в Linux

## 9. Заключение

Процесите са основата на работата на операционната система. Без тях компютърът може да работи само с една програма наведнъж. Управлението на процесите е ключова задача на ОС.

### Запомни:



- **Процес** - програма в изпълнение
- **Петте състояния** - Нов, Готов, Работи, Чака, Завършен
- **PCB** - съхранява информация за процеса
- **Context Switch** - превключване между процеси
- **Многозадачност** - много процеси "едновременно"
- **Планиране** - решаване кой процес да работи
- **Йерархия** - процесите образуват дърво