

Процеси. Пакетна обработка, многозадачност, времееделение — 126

1. Какво е процес

Процесът е програма в изпълнение. Това е активна инстанция на програмен код, която се изпълнява от операционната система и притежава собствени ресурси.

Разлика между програма и процес:

- **Програма** - статичен файл на диска, съдържащ инструкции
- **Процес** - динамична изпълняваща се програма с ресурси (памет, процесорно време, файлове)

Пример:

Имате инсталиран браузър Firefox (програма). Когато го стартирате, се създава процес Firefox, който използва процесор и памет. Ако отворите Firefox два пъти, ще имате два отделни процеса.

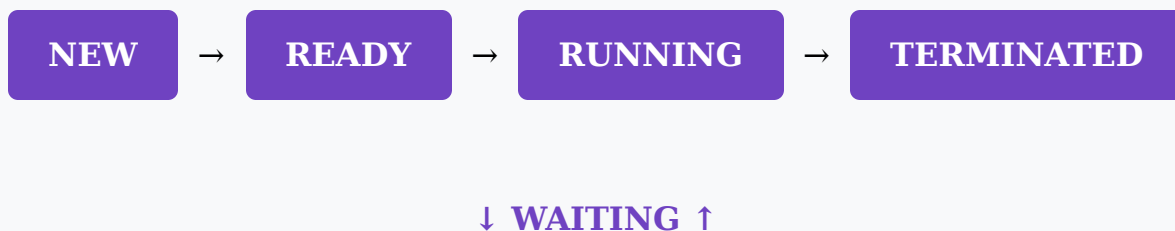
2. Компоненти на процес

Всеки процес съдържа:

- **Програмен код (Text Section)** - изпълнимите инструкции
- **Програмен брояч (Program Counter)** - адрес на следващата инструкция
- **Регистри на процесора** - временни данни
- **Стек (Stack)** - временни данни, функции, локални променливи
- **Данни (Data Section)** - глобални променливи
- **Heap** - динамично заделена памет

3. Състояния на процес

Процесът преминава през различни състояния по време на своя жизнен цикъл:



Състояние	Описание
NEW	Процесът се създава
READY	Процесът чака на опашка за процесорно време
RUNNING	Процесът се изпълнява на процесора
WAITING	Процесът чака събитие (I/O операция, сигнал)
TERMINATED	Процесът приключва изпълнението си

4. Пакетна обработка (Batch Processing)

Най-ранната форма на многопрограмно изпълнение, където задачите се изпълняват последователно без взаимодействие с потребителя.

Характеристики:

- **Групиране** - задачите се събират в пакети
- **Последователност** - една задача приключва преди да започне следващата
- **Без интерактивност** - няма взаимодействие по време на изпълнение
- **Ефективност** - максимална употреба на процесора

Примери за пакетна обработка:

- Заплащане на месечни заплати в компания
- Обработка на банкови транзакции в края на деня
- Генериране на статистически отчети
- Автоматични резервни копия (backup)

5. Многозадачност (Multitasking)

Способността на операционната система да изпълнява множество процеси едновременно чрез бързо превключване между тях.

Типове многозадачност:

- **Кооперативна (Cooperative)** - процесите сами решават кога да отстъпят процесора
- **Превантивна (Preemptive)** - операционната система принудително взема процесора от процесите

Предимства на многозадачността:

- **По-добра употреба на ресурсите** - процесорът не стои празен
- **Подобрена реактивност** - системата отговаря на действията на потребителя
- **Паралелизъм** - множество задачи напредват едновременно

6. Времеделене (Time Sharing)

Техника, при която процесорното време се разделя на малки интервали (квант време), и всеки процес получава достъп до процесора за кратък период.

Как работи:

1. Операционната система дава на процес 1 квант време (напр. 10ms)
2. Процесът се изпълнява за този период
3. След изтичане на времето процесът се прекъсва
4. Друг процес получава процесора
5. Цикълът се повтаря

Пример с 3 процеса:

Процес А → 10ms → Процес В → 10ms → Процес С → 10ms → Процес А → ...

Резултат: Всички три процеса изглеждат като че се изпълняват едновременно

Характеристики:

- **Интерактивност** - потребителят може да работи с няколко програми
- **Бързо превключване** - за човека изглежда като паралелизъм
- **Справедливост** - всеки процес получава време

7. Създаване на процес в Windows

В Windows процесите се създават с функцията `CreateProcess()` на ниво API. На потребителско ниво процесите се стартират чрез:

- **Start Menu** - двоен клик върху програма
- **Command Prompt / PowerShell** - изпълнение на команди
- **Task Scheduler** - автоматично стартиране
- **Друга програма** - родителски процес създава дъщерен

Стартиране на процес от Command Prompt:

```
C:\> notepad.exe  
C:\> calc.exe  
C:\> start chrome.exe  
C:\> start /B program.exe
```

```
C:\> tasklist | findstr notepad
notepad.exe      5432 Console      1      12,345 K
```

Стартиране в background:

```
C:\> start /B my_program.exe
```

```
C:\> start "" "C:\Program Files\My App\app.exe"
```

- **/B** - стартира процеса без нов прозорец
- **start** - стартира програма в отделен процес

8. Унищожаване на процес в Windows

Процесът може да завърши нормално или да бъде прекратен принудително.

Начини за приключване:

- **Нормално завършване** - процесът изпълнява последната си инструкция и излиза
- **Грешка** - програмна грешка (деление на нула, недостъпна памет)
- **Фатална грешка** - неочаквано събитие (crash)
- **Принудително** - от потребител или друг процес

Метод 1: Task Manager (графичен интерфейс)

Стъпки:

1. Натиснете **Ctrl + Shift + Esc**
2. Отворете таб **Processes**
3. Намерете процеса в списъка
4. Кликнете с десен бутон върху него
5. Изберете **End Task**

Допълнителни опции в Task Manager:

- **End Task** - меко прекратяване (препоръчително)
- **End Process Tree** - прекратява процеса и всичките му дъщерни процеси
- **Details → End Process** - принудително прекратяване

Метод 2: Command Prompt (команден ред)

Основни команди:

```
C:\> tasklist
Image Name                PID Session Name        Session#    Mem
Usage
firefox.exe               1234 Console                1          256,789
K
chrome.exe                5678 Console                1          512,345
K

C:\> tasklist | findstr firefox
firefox.exe               1234 Console                1          256,789
K

C:\> taskkill /PID 1234
SUCCESS: Sent termination signal to the process with PID 1234.

C:\> taskkill /F /PID 1234
SUCCESS: The process with PID 1234 has been terminated.

C:\> taskkill /IM firefox.exe
```

```
SUCCESS: Sent termination signal to the process "firefox.exe" with PID 1234.
```

```
C:\> taskkill /F /IM firefox.exe
```

```
SUCCESS: The process "firefox.exe" with PID 1234 has been terminated.
```

Параметри на **taskkill**:

- **/PID [номер]** - прекратява процес по Process ID
- **/IM [име.exe]** - прекратява процес по име на програмата
- **/F** - принудително прекратяване (Force)
- **/T** - прекратява процеса и всички дъщерни процеси

Метод 3: PowerShell

```
PS C:\> Get-Process
```

Handles	NPM(K)	PM(K)	WS(K)	CPU(s)	Id	SI	ProcessName
-----	-----	-----	-----	-----	--	--	-----
456	23	45678	67890	12.34	1234	1	firefox

```
PS C:\> Get-Process firefox
```

Handles	NPM(K)	PM(K)	WS(K)	CPU(s)	Id	SI	ProcessName
-----	-----	-----	-----	-----	--	--	-----
456	23	45678	67890	12.34	1234	1	firefox

```
PS C:\> Stop-Process -Id 1234
```

```
PS C:\> Stop-Process -Name firefox
```

```
PS C:\> Stop-Process -Id 1234 -Force
```

```
PS C:\> Get-Process firefox | Stop-Process
```


Кога да използваме принудително прекратяване:

- Програмата не отговаря (висна)
- Процесът не реагира на нормално затваряне
- Системата е блокирана от процеса

Внимание: Принудителното прекратяване може да доведе до загуба на незапазени данни!

9. Преглед на процеси в Windows

Метод 1: Task Manager

Стартиране:

- `Ctrl + Shift + Esc` - пряко отваряне
- `Ctrl + Alt + Del` → Task Manager
- Десен бутон на taskbar → Task Manager

Информация в Task Manager:

Колона	Описание
Name	Име на процеса/програмата
PID	Process ID - уникален номер
Status	Running, Suspended, Not Responding
CPU %	Процент употреба на процесора

Колона	Описание
Memory	Заета оперативна памет
Disk	Употреба на диска
Network	Мрежова активност

Полезни табове в Task Manager:

- **Processes** - всички текущи процеси с консумация на ресурси
- **Performance** - графики за CPU, Memory, Disk, Network
- **Details** - детайлна информация за процесите (PID, User, Threads)
- **Services** - системни услуги
- **Startup** - програми при стартиране

Метод 2: Command Prompt - команда `tasklist`

```
C:\> tasklist
```

Image Name	PID	Session	Name	Session#	Mem
Usage					
System Idle Process	0	Services		0	8
K					
System	4	Services		0	1,234
K					
smss.exe	256	Services		0	456
K					
csrss.exe	384	Services		0	3,456
K					
wininit.exe	476	Services		0	2,345
K					
explorer.exe	1234	Console		1	67,890
K					
firefox.exe	5678	Console		1	256,789
K					

```
chrome.exe          9012 Console          1    512,345
K

C:\> tasklist | findstr firefox
firefox.exe        5678 Console          1    256,789
K

C:\> tasklist /V
Image Name          PID Status             User Name          CPU Time Window Title
firefox.exe        5678 Running           PC\User            0:05:23 Mozilla
Firefox

C:\> tasklist /FI "MEMUSAGE gt 100000"
Image Name          PID Session Name      Session#    Mem
Usage
chrome.exe          9012 Console          1    512,345
K
firefox.exe        5678 Console          1    256,789
K
```

Параметри на **tasklist**:

- **/V** - подробна информация (verbose)
- **/FI "filter"** - филтриране на процеси
- **/SVC** - показва услугите за всеки процес
- **/M** - показва заредените DLL модули

Метод 3: PowerShell - **Get-Process**

```
PS C:\> Get-Process
Handles  NPM(K)  PM(K)  WS(K)  CPU(s)  Id  SI ProcessName
-----  -
145      9       2456   5678   0.12    384  0 csrss
567      34      67890  123456 12.34   1234  1 explorer
456      23      45678  67890  23.45   5678  1 firefox
789      45     234567 456789 45.67   9012  1 chrome
```

```

PS C:\> Get-Process firefox
Handles    NPM(K)    PM(K)      WS(K)      CPU(s)      Id  SI ProcessName
-----
456        23        45678      67890      23.45      5678 1 firefox

PS C:\> Get-Process | Sort-Object CPU -Descending | Select-Object -First 5
Handles    NPM(K)    PM(K)      WS(K)      CPU(s)      Id  SI ProcessName
-----
789        45        234567    456789    45.67      9012 1 chrome
456        23        45678      67890      23.45      5678 1 firefox
567        34        67890     123456     12.34     1234 1 explorer

PS C:\> Get-Process | Where-Object {$_.CPU -gt 10}

PS C:\> Get-Process -Id 5678 | Format-List *
```

Полезни PowerShell команди:

- `Get-Process | Sort-Object CPU -Descending` - сортиране по CPU
- `Get-Process | Sort-Object WS -Descending` - сортиране по памет
- `Get-Process | Export-Csv processes.csv` - експорт в CSV файл

10. Практически упражнения

Упражнение 1: Преглед на процеси с Task Manager

1. Отворете Task Manager с `Ctrl + Shift + Esc`
2. Идентифицирайте процеса с най-голяма употреба на CPU
3. Намерете PID на вашия браузър
4. Проверете колко памет използва
5. Отворете таб **Details** и намерете процеса по PID

Упражнение 2: Работа с Command Prompt

1. Отворете Command Prompt
2. Изпълнете `tasklist` за преглед на всички процеси
3. Стартирайте Notepad с команда `notepad.exe`
4. Намерете PID на Notepad с `tasklist | findstr notepad`
5. Прекратете го с `taskkill /IM notepad.exe`
6. Проверете дали е премахнат със `tasklist | findstr notepad`

Упражнение 3: PowerShell команди

1. Отворете PowerShell
2. Изпълнете `Get-Process` за преглед на процесите
3. Намерете процесите с най-голяма CPU употреба с:
`Get-Process | Sort-Object CPU -Descending | Select-Object -First 5`
4. Намерете процесите с най-голяма памет с:
`Get-Process | Sort-Object WS -Descending | Select-Object -First 5`
5. Експортирайте списъка в CSV файл с:
`Get-Process | Export-Csv processes.csv`

Упражнение 4: Мониторинг и анализ

1. Стартирайте няколко програми (браузър, Calculator, Notepad)
2. Отворете Task Manager и наблюдавайте промените в употребата на ресурси
3. Кликнете на колоната **CPU** за сортиране

4. Отворете таб **Performance** и наблюдавайте графиките
5. В таб **Details** намерете дъщерни процеси (проверете User Name колоната)

Упражнение 5: Филтриране на процеси

1. Отворете Command Prompt
2. Покажете само процесите, които използват над 100 MB памет:

```
tasklist /FI "MEMUSAGE gt 100000"
```

3. Покажете всички процеси със статус "Running":

```
tasklist /FI "STATUS eq running"
```

4. Покажете детайлна информация за конкретен процес:

```
tasklist /V /FI "IMAGENAME eq chrome.exe"
```

11. Заключение

Процесите са основната единица за изпълнение в операционните системи. Разбирането на жизнения им цикъл, състояния и управление е критично за ефективна работа с Windows.

Ключови точки:

- **Процес** - програма в изпълнение с ресурси и състояния
- **Пакетна обработка** - последователно изпълнение без интерактивност
- **Многозадачност** - паралелно изпълнение на множество процеси
- **Времеделене** - разделяне на процесорно време на квантове
- **Състояния** - NEW, READY, RUNNING, WAITING, TERMINATED
- **Инструменти за управление:**

- Task Manager - графичен интерфейс
- tasklist, taskkill - Command Prompt
- Get-Process, Stop-Process - PowerShell

Полезни клавишни комбинации:

- `Ctrl + Shift + Esc` - Task Manager
- `Ctrl + Alt + Del` - Security menu
- `Alt + F4` - затваряне на активна програма
- `Windows + X` → Task Manager