

Алгоритми за управление на паметта: First Fit, Best Fit, Worst Fit

1. Въведение в управлението на паметта

- **Цел:** Разпределяне на ограничена памет между процеси в компютърната система.
 - **Проблем:** Неефективно разпределяне на паметта може да доведе до фрагментация.
 - **Алгоритми:** **First Fit**, **Best Fit** и **Worst Fit** са три основни стратегии за разпределение на паметта.
-

2. Алгоритъм First Fit

- **Описание:** Разпределя памет в първия блок, който е достатъчно голям за процеса.

Процес:

1. Сканира списъка със свободни блокове от началото.
2. Избира първия блок, който е достатъчно голям.
3. Ако блокът е по-голям от необходимото, остатъчната част остава свободна.

- **Пример:**
 - Свободни блокове: 50, 30, 70, 20
 - Процес с нужда от 40 единици памет
 - **First Fit** ще избере блок с размер 50 (първият подходящ).
 - **Предимства:**
 - Бърз и лесен за имплементиране.
 - Спира веднага, когато намери подходящ блок.
 - **Недостатъци:**
 - Може да доведе до **вътрешна фрагментация**, тъй като блокът може да е по-голям от необходимото.
 - Не оптимизира разпределението на паметта.
-

3. Алгоритъм Best Fit

- **Описание:** Избира най-малкия свободен блок, който може да побере процеса.

Процес:

1. Сканира всички свободни блокове.
2. Избира най-малкия блок, който е достатъчно голям.
3. Остатъчната част от блока се връща в списъка със свободни блокове.

- **Пример:**

- Свободни блокове: 50, 30, 70, 20
- Процес с нужда от 40 единици памет
- **Best Fit** ще избере блок с размер 50 (най-малък, който е подходящ).

- **Предимства:**

- Намалява **вътрешната фрагментация**.
- По-добро използване на паметта в дългосрочен план.

- **Недостатъци:**

- По-бавен, тъй като сканира всички блокове.
- Може да доведе до **външна фрагментация**, оставяйки малки, неизползваеми блокове.

4. Алгоритъм Worst Fit

- **Описание:** Избира най-големия свободен блок за разпределяне на паметта.

Процес:

1. Сканира всички свободни блокове.
2. Избира най-големия блок, който е достатъчно голям.
3. Остава остатъчната част от блока свободна.

- **Пример:**

- Свободни блокове: 50, 30, 70, 20
- Процес с нужда от 40 единици памет
- **Worst Fit** ще избере блок с размер 70 (най-голям).

- **Предимства:**

- Използва по-големи блокове, което може да намали вътрешната фрагментация.
 - Оставя по-големи свободни блокове за бъдещи процеси.
- **Недостатъци:**
 - Може да доведе до **външна фрагментация**.
 - По-големите блокове могат да се разделят на малки парчета, които не могат да бъдат използвани ефективно.
-

5. Обобщение на алгоритмите

- **First Fit:**
 - Лесен за имплементиране.
 - Може да доведе до вътрешна фрагментация.
 - **Бърз**, но не оптимален в дългосрочен план.
- **Best Fit:**
 - Оптимизира използването на паметта.
 - Може да води до външна фрагментация.
 - **По-бавен**, тъй като сканира всички блокове.
- **Worst Fit:**
 - Намалява вътрешната фрагментация.
 - Може да доведе до голяма външна фрагментация.
 - Избира най-големите блокове за процесите.