

☐ 1

Internal Fragmentation: Memori yang dialokasikan lebih besar dibandingkan memori yang diperlukan dan digunakan oleh proses.

☐

External Fragmentation: Selisih jarak antar memori<sup>2</sup> yang telah dialokasikan.

☐☐

OS akan melakukan berbagai teknik memory management untuk meningkatkan efisiensi alokasi memori.

☐☐☐

Teknik<sup>3</sup>:

☐

- Partitioning: Membagi memori dalam bentuk partisi secara Fixed / Dynamic saat dibutuhkan.

☐☐☐

- Paging: Membagi memori dalam ukuran yang sama sesuai dengan ukuran Frame. kemudian proses akan menggunakan hasil pembagian tersebut

☐☐☐☐

- Segmentation: Membagi memori dalam ~~ukuran~~ bentuk segmentasi yang disimpan ke lokasi memori yang berbeda

☐☐☐

- Buddy System: Membagi memori dengan bentuk dinamik partisi sebesar pangkat 2.

☐☐☐☐ 2

Virtual Memory: Teknik alokasi storage dengan menggunakan Secondary memori sebagai main memori

☐☐☐

Teknik ini dapat membantu menyimpan data yang jarang diakses ke dalam virtual memori sehingga memori Fisik/main akan tersedia lebih banyak sebagai tempat untuk digunakan oleh data<sup>2</sup> yang lebih sering dipakai.

☐☐☐



3 Shared Memori : Teknik untuk membuat alur komunikasi antar proses sehingga antar proses dapat saling mengubah/bertukar data satu sama lain.

Contoh : Dalam proses forking, child process dapat berbagi memori dengan parent sehingga dapat mempermudah/efisien mengelola memori

4 ex:  $T_2$  = Halaman urutan

FFO (First in First Out)

Urutan 2 5 7 3 5 5 2 4 1 6

~~Frame 2 2 2 3 3 3 3 3 3 1 1~~  
~~5 5 5 5 5 5 5 5 5 1~~  
~~7 7 7 7 2 2 2 2 2 2~~

Frame 2 2 2 3 3 3 3 3 3 1 1  
 5 5 5 5 5 5 2 2 2 6  
 7 7 7 7 7 4 4 4 4

Optimal

Urutan 2 5 7 3 5 5 2 4 1 6

Frame 2 2 2 2 2 2 2 4 1 6  
 5 5 5 5 5 5 5 5 5 5  
 7 3 3 3 3 3 3 3 3 3



LRU:

Urutan: 2 5 7 3 5 5 2 4 1 6

Frame: 2 2 2 3 3 3 3 4 4 4

5 5 5 5 5 5 5 5 1 1

7 7 7 7 2 2 2 6

Clock:

Urutan: 2 5 7 3 5 5 2 4 1 6

Frame: 2<sup>\*</sup> 2<sup>\*</sup> → 2<sup>\*</sup> 3<sup>\*</sup> 3<sup>\*</sup> 3<sup>\*</sup> → 3<sup>\*</sup> 3<sup>\*</sup> → 3 6<sup>\*</sup>→ 5<sup>\*</sup> 5<sup>\*</sup> → 5 → 5<sup>\*</sup> → 5<sup>\*</sup> 5 4<sup>\*</sup> 4<sup>\*</sup> → 4<sup>\*</sup>→ 7<sup>\*</sup> 7 7 7 2<sup>\*</sup> → 2 1<sup>\*</sup> 1<sup>\*</sup>

5 Teknik<sup>2</sup> tersebut digunakan oleh OS ketika halaman yang diperlukan tidak ada di memori (~~Page~~ Fault)

Contoh

Penggunaan web tab browser, ketika new tab akan dimuat, OS akan mengecek apakah tab tersebut terdapat pada memori. Jika tidak ada di memori, OS akan menjalankan algoritma tersebut.