

# Sistem Operasi

Managemen Memori

2016

# Memory Management

- Pada uniprogramming, memori terbagi menjadi 2
  - Milik OS (kernel, resident monitor)
  - Program yang sedang di eksekusi
- Pada multiprogramming, bagian user dapat dinyatakan sebagai bagian dari multiple process. Pekerjaan dari subdivisi ini dilakukan secara dinamis oleh OS. Inilah yang disebut **manajemen memori**.

# Memory Management

- Tujuan utama manajemen memori, untuk memenuhi :
  - Relocation
  - Protection
  - Sharing
  - Logical organization
  - Physical organization

# Relocation

- Pada sistem multiprogramming, memory utama pada umumnya berbagi antara beberapa proses.
- Biasanya, tidak mungkin programmer mengetahui keuntungan dari sebuah program yang berdiam di memori utama dalam waktu tertentu.
- Maka dari itu harus ada kemampuan melakukan swap proses yang aktif untuk memaksimalkan utilitas prosesor.
- Maka dari itu perlu dilakukan **RELOKASI** proses ke area memori yang lain.

# Protection

- Setiap proses harus dapat diproteksi dari interferensi yang tidak diinginkan, yang berasal dari proses lainnya, baik secara tidak sengaja maupun disengaja.
- Proses-proses tidak dapat melakukan referensi lokasi memori yang bertujuan untuk membaca atau menulis ke dalam memori tanpa izin
- Hal ini terjadi karena lokasi memori sebuah program dalam memori utama tidak dapat ditebak, dan mustahil melakukan pengecekan alamat pada saat waktu kompilasi.

# Sharing

- Mekanisme proteksi harus memiliki fleksibilitas untuk mengizinkan beberapa proses untuk mengakses memori utama.
- Sebagai contoh, jika beberapa proses mengeksekusi program yang sama, akan menjadi sebuah keuntungan jika masing-masing proses dapat mengakses program yang sama.

# Physical Organization

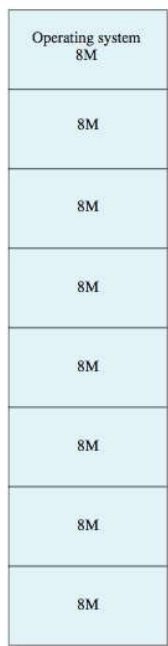
- Struktur fisik memori biasanya terbagi menjadi 2
  - Main memory (fast access, high cost, volatile)
  - Secondary memory (slower, cheaper, non volatile)
- Hal ini adalah hal yang sangat wajib untuk diperhatikan.
  - Program ditambah datanya dapat menghabiskan resource yang tersedia pada memori utama. Terdapat teknik **overlaying**, dimana program dan data diatur dan disimpan dalam area memori yang sama.
  - Pada sistem multiprograming, programmer pada saat membuat program tidak dapat mengetahui sebesar apa data yang akan digunakan.

# Partisi Memori

- Fixed Partitioning
- Dynamic Partitioning
- Simple paging
- Simple Segmentation
- Virtual Memory Paging
- Virtual Memory Segmentation



# Fixed Partitioning



(a) Equal-size partitions



(b) Unequal-size partitions

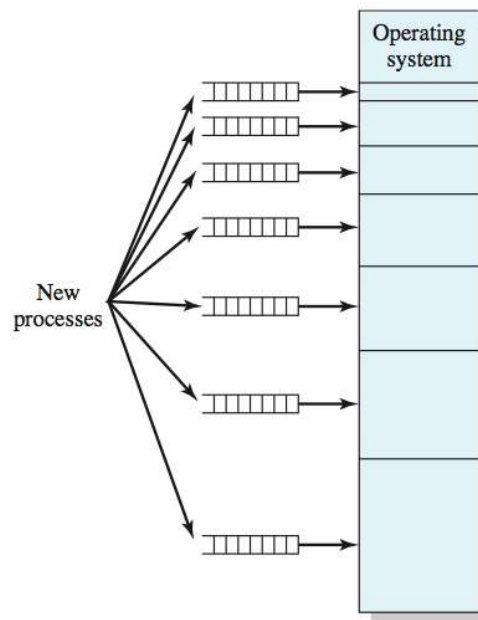
- Memori dibagi ke beberapa bagian, ada yang ukurannya sama, ada juga yang tidak.
- Jika terdapat sebuah proses yang membutuhkan memori, dapat mengisi slot yang kosong.

# Fixed Partitioning

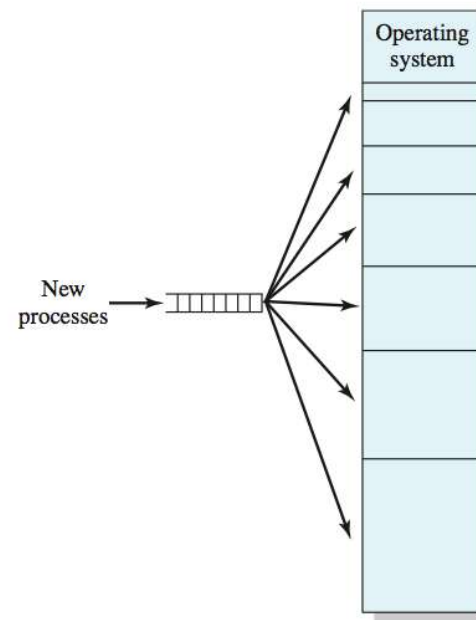
Terdapat 2 masalah pada partisi jenis ini.

1. Apa yang terjadi jika sebuah program ternyata membutuhkan memori lebih besar dari partisi memori? Jika harus terjadi, maka akan terjadi overlaying pada program dan data
2. Tidak efisien. Bayangkan jika slot yang ada hanya digunakan 1/10nya.

# Placement Algorithm

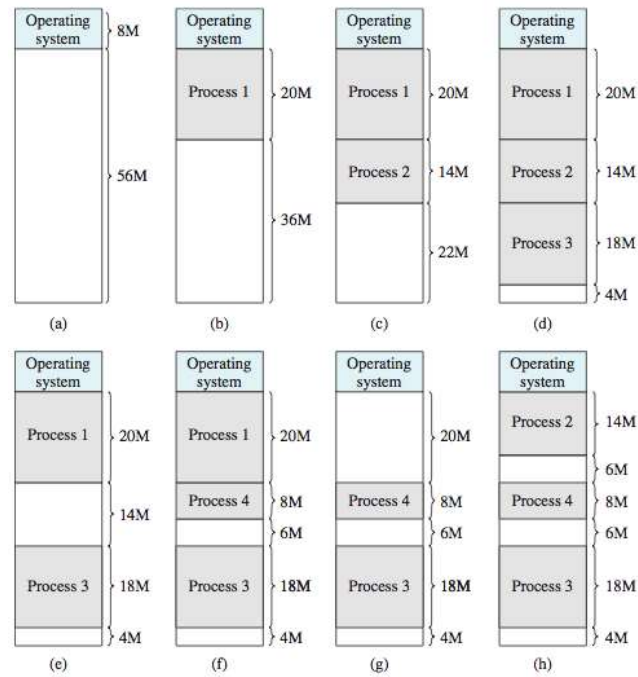


(a) One process queue per partition

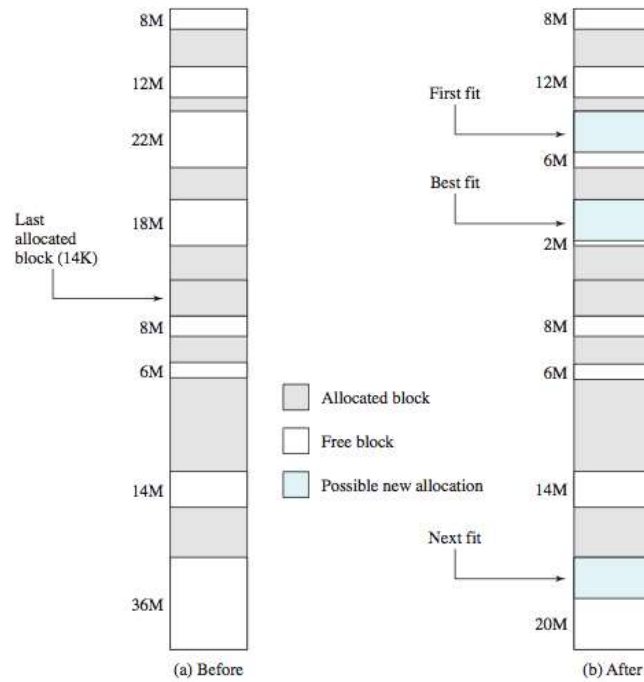


(b) Single queue

# Dynamic Partitioning

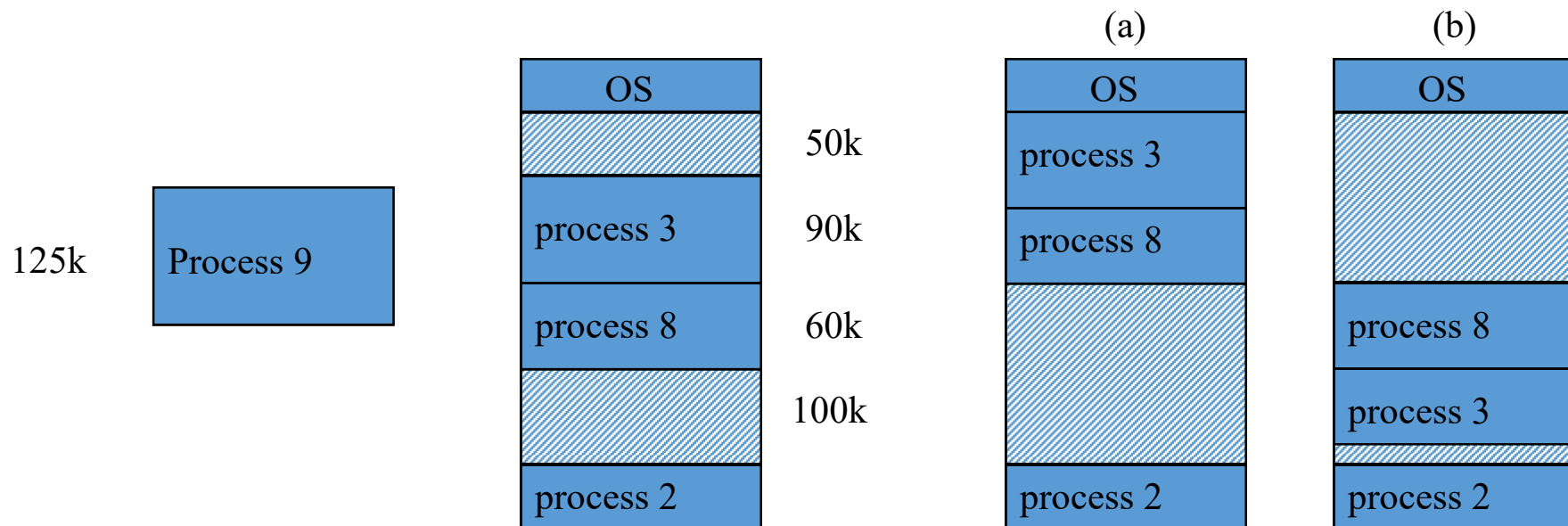


# Placement Algorithm

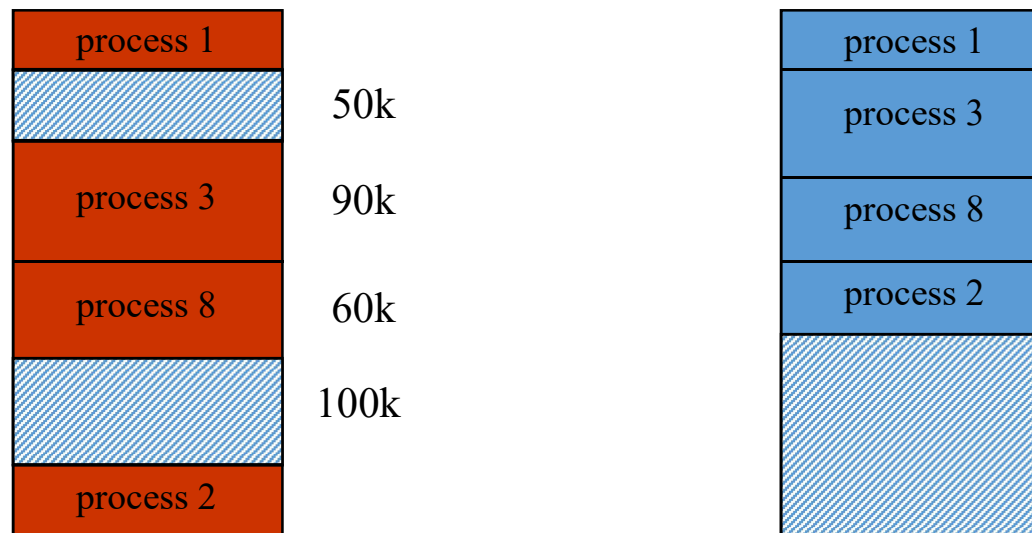


# Memampatan (compaction)

- Susun isi memori untuk mendapatkan blok kosong yang cukup besar
- Hanya jika relokasi dinamis!



# Biaya Pemampatan



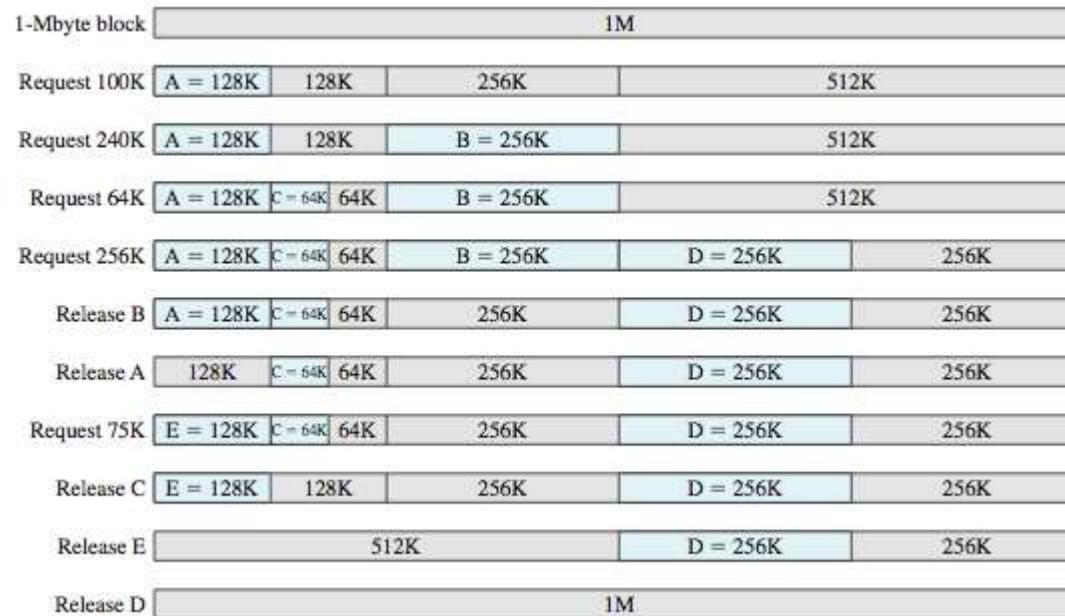
- 128 MB RAM, 100 nsec/akses
  - ➔ 1.5 detik untuk pemampatan!
- Disk lebih lambat!

# Buddy System

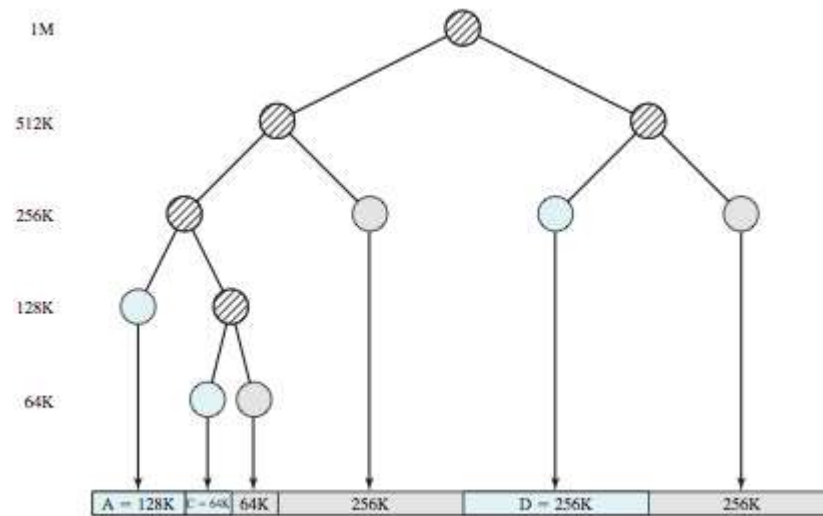
- 1-Mbyte initial block.
  - The first request, A, is for 100 Kbytes, for which a 128K block is needed.
  - The initial block is divided into two 512K buddies.
  - The first of these is divided into two 256K buddies, and ...
  - the first of these is divided into two 128K buddies, one of which is allocated to A.
  - The next request, B, requires a 256K block. Such a block is already available and is allocated. The process continues with splitting and coalescing occurring as needed.
  - Note that when E is released, two 128K buddies are coalesced into a 256K block, which is immediately coalesced with its buddy.



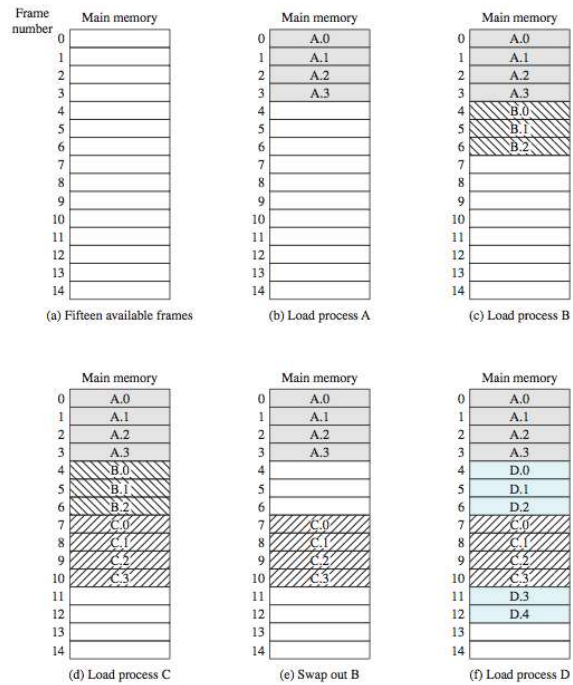
# Buddy System



# Buddy System



# Paging



# Data Structure Page Table

0	0
1	1
2	2
3	3

Process A  
page table

0	—
1	—
2	—

Process B  
page table

0	7
1	8
2	9
3	10

Process C  
page table

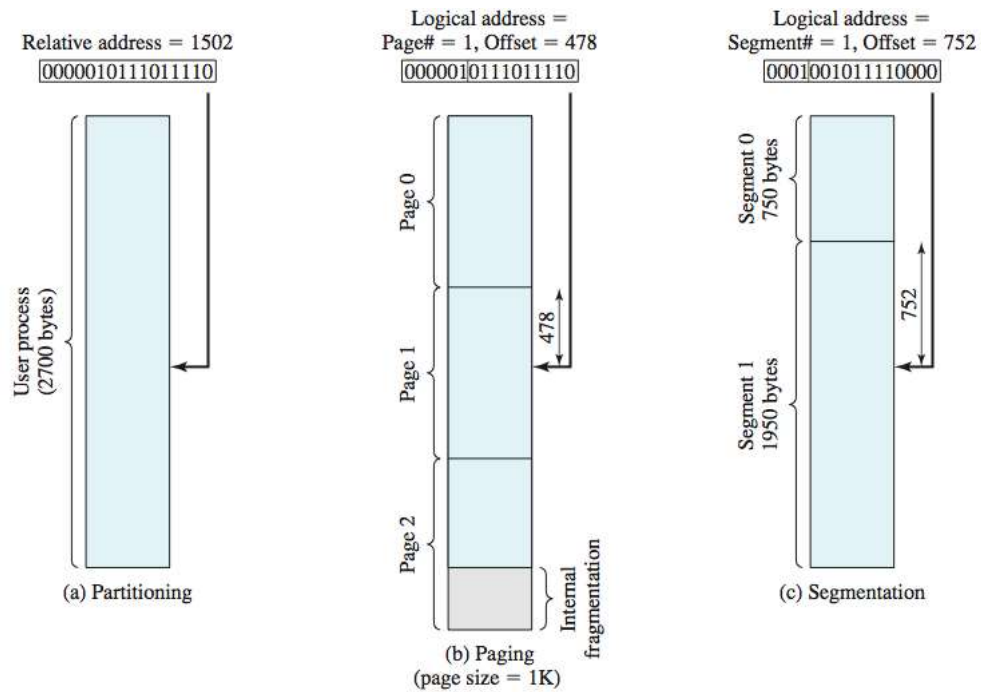
0	4
1	5
2	6
3	11
4	12

Process D  
page table

13
14

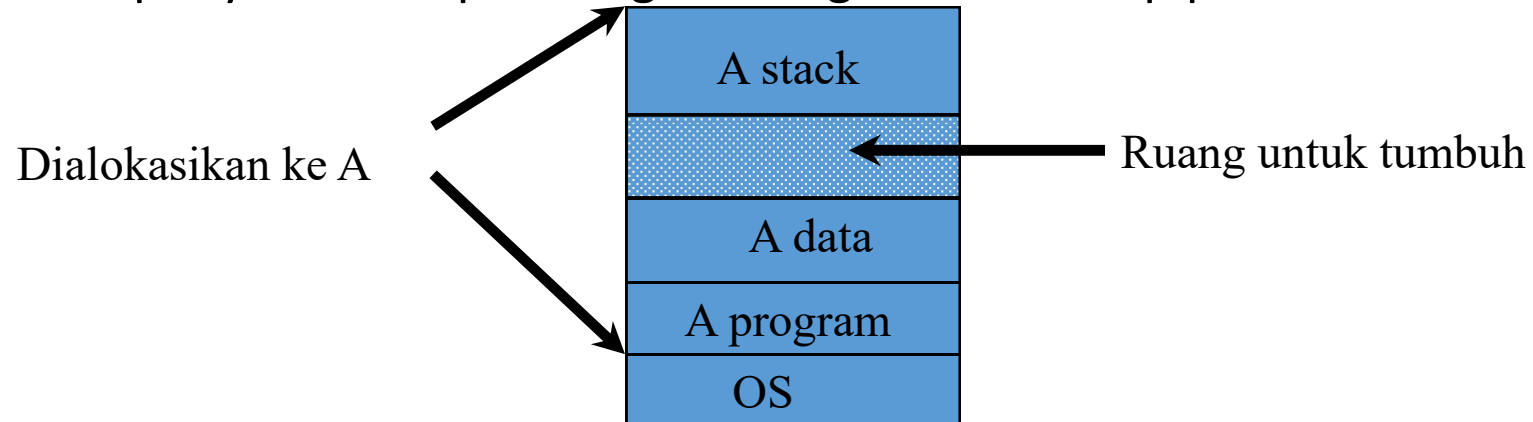
Free frame  
list

# Segmentation



# Fragmentasi Internal

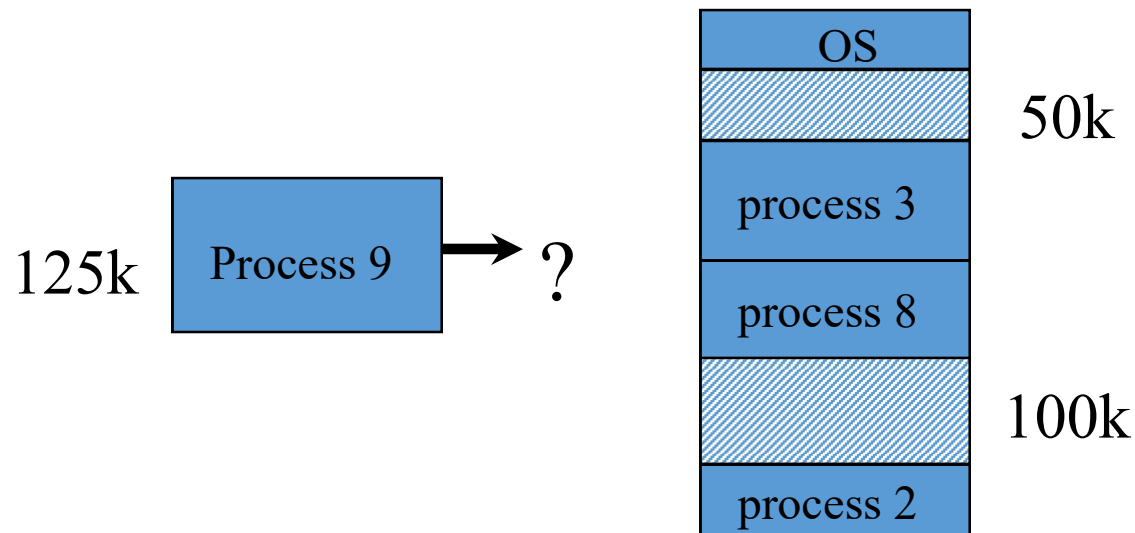
- Mempunyai beberapa ruang “kosong” untuk setiap proses



- Fragmentasi Internal – memori yang dialokasikan mungkin sedikit lebih besar dari memori yang diminta dan tidak digunakan.

# Fragmentasi Eksternal

- Fragmentasi Eksternal – ruang memori total yang terdapat untuk memenuhi permintaan tetapi tidak kontinu



# Pertanyaan Review

- Apa yang disebut partisi ukuran tetap? Fragmentasi apa yang dipunyaiinya?
- Apa yang disebut partisi ukuran variabel? Fragmentasi apa yang dipunyaiinya?
- Apa yang disebut pemampatan, kenapa diperlukan, dan apa kekurangannya?
- Apa yang disebut paging? Fragmentasi apa yang dipunyaiinya?