

# Virtual Memory

Pertemuan 11

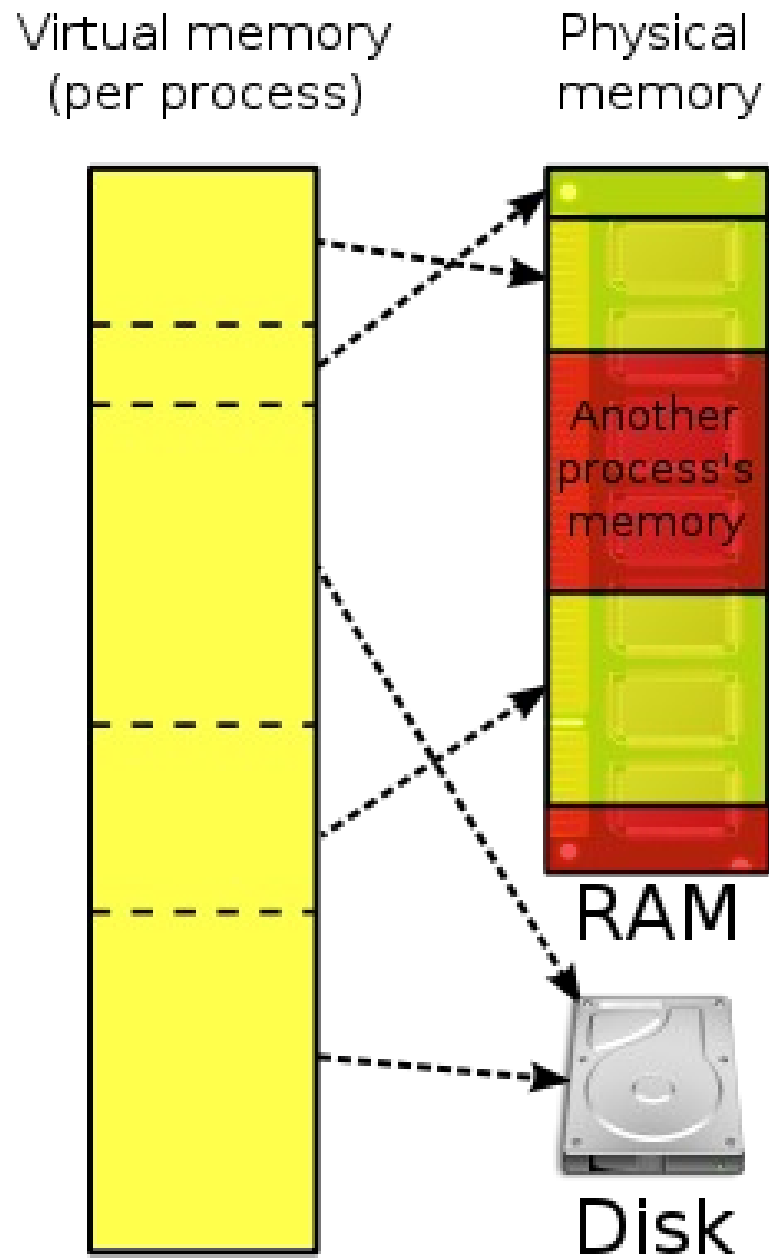
# Definisi

- Sebuah teknik memori manajemen yang menyediakan ilusi kepada pengguna seakan punya memori komputer besar
- Namun faktanya adalah penyimpanan sekunder digunakan untuk menyimpan proses-proses lain

# Lanjutan

- Virtual Memori adalah skema alokasi penyimpanan yang terletak di memori sekunder yang dapat dialamatkan/ditunjuk seolah bagian dari memori utama

# Ilustrasi



# Lanjutan

- Virtual Memori adalah bagian dari komputer modern sehingga memerlukan dukungan perangkat khusus di dalam CPU
- Diperkenalkan pertama kali oleh mode terlindungi dari Intel 80286

# Lanjutan

- Ukuran dari penyimpanan Memori Virtual dibatasi oleh skema pengalamatan sistem dan jumlah memory yang ada

# Page dan Frame

- Page/Virtual Page/Memory Page adalah sebuah blok berkesinambungan dengan ukuran yang tetap dari Virtual Memory
- Page Frame adalah sebuah blok dengan ukuran terkecil yang berkesinambungan dari Physical Memory

# Ukuran Page per Arsitektur

Architecture	Smallest page size	Larger page sizes
32-bit x86 <sup>[18]</sup>	4 KIB	4 MIB in PSE mode, 2 MIB in PAE mode <sup>[19]</sup>
x86-64 <sup>[18]</sup>	4 KIB	2 MIB, 1 GiB (only when the CPU has PDPE1GB flag)
IA-64 (Itanium) <sup>[20]</sup>	4 KIB	8 KIB, 64 KIB, 256 KIB, 1 MIB, 4 MIB, 16 MIB, 256 MIB <sup>[19]</sup>
Power ISA <sup>[21]</sup>	4 KIB	64 KIB, 16 MIB, 16 GiB
SPARC v8 with SPARC Reference MMU <sup>[22]</sup>	4 KIB	256 KIB, 16 MIB
UltraSPARC Architecture 2007 <sup>[23]</sup>	8 KIB	64 KIB, 512 KIB (optional), 4 MIB, 32 MIB (optional), 256 MIB (optional)
ARMv7 <sup>[24]</sup>	4 KIB	64 KIB, 1 MIB ("section"), 16 MIB ("supersection") (defined by a page size)



# Teknik Virtual Memory

- Semua memori yang dialamatkan dari sebuah proses adalah alamat logikal yang secara dinamis diubah ke alamat fisik dalam satu waktu
- Sebuah proses bisa jadi dipecah menjadi beberapa potongan, dan sebagian menuju memori kedua

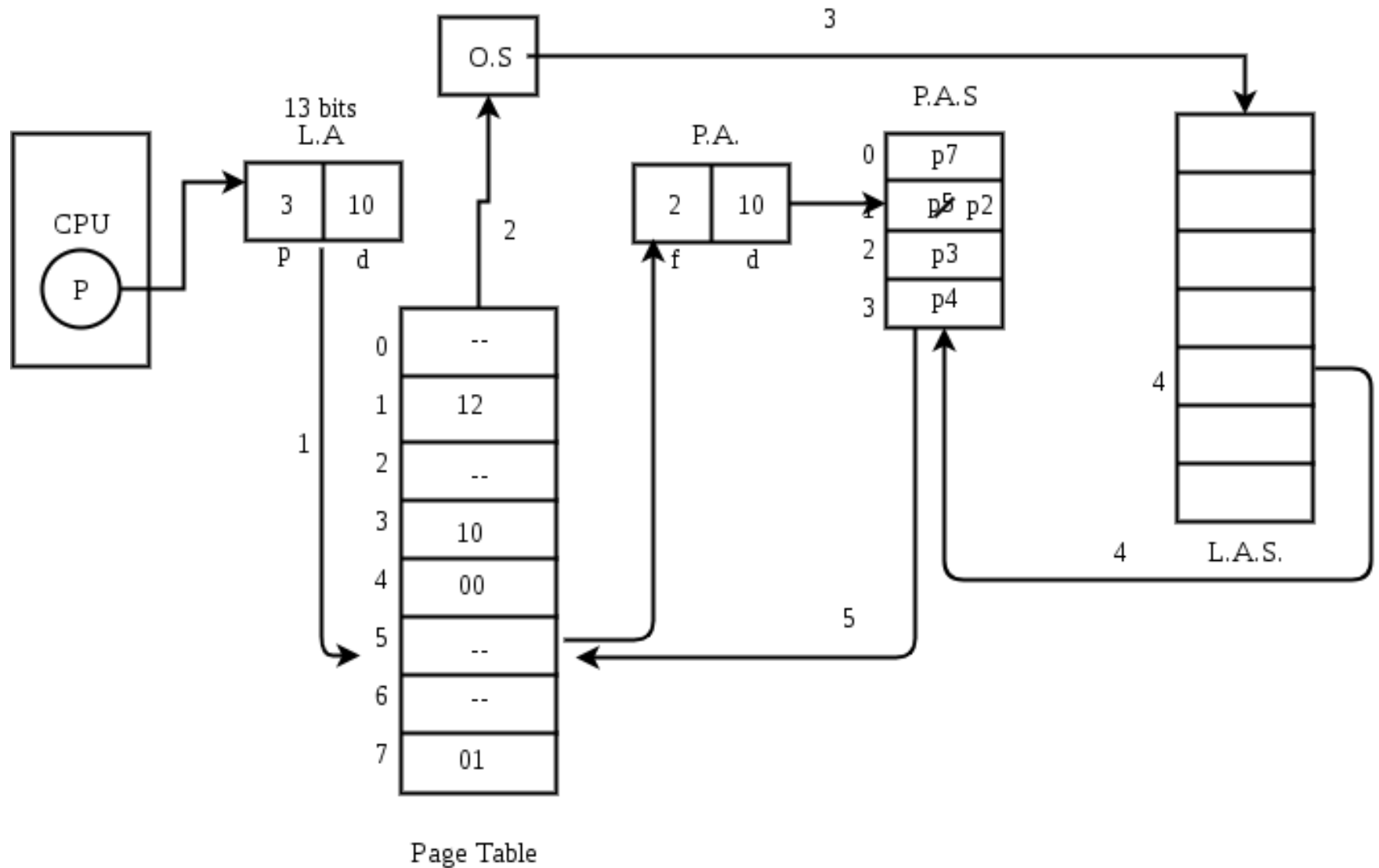
# Implementasi VM

- Virtual Memori diimplementasikan dengan dua cara
  - Demand Paging
  - Demand Segmentation

# Demand Paging

- Proses dari memuat page ke memori secara diperintah (on demand) ketika page fault error terjadi

# Ilustrasi



# Lanjutan

- Jika CPU coba menunjuk page yang tidak ada di memory maka dia akan membuat interupsi akan memory access fault
- Proses yang terinterupsi akan masuk ke state Blocking. Jika ingin berlanjut maka proses harus membawa page yang hilang

# Lanjutan

- SO akan mencari page yang dibutuhkan di logical address space
- Page yang dibutuhkan akan dibawa dari logical address space ke physical address space
- Tabel Page akan diperbarui
- Sinyal akan dikirimkan ke CPU untuk melanjutkan eksekusi program, dan melanjutkannya ke state Ready

# Keuntungan

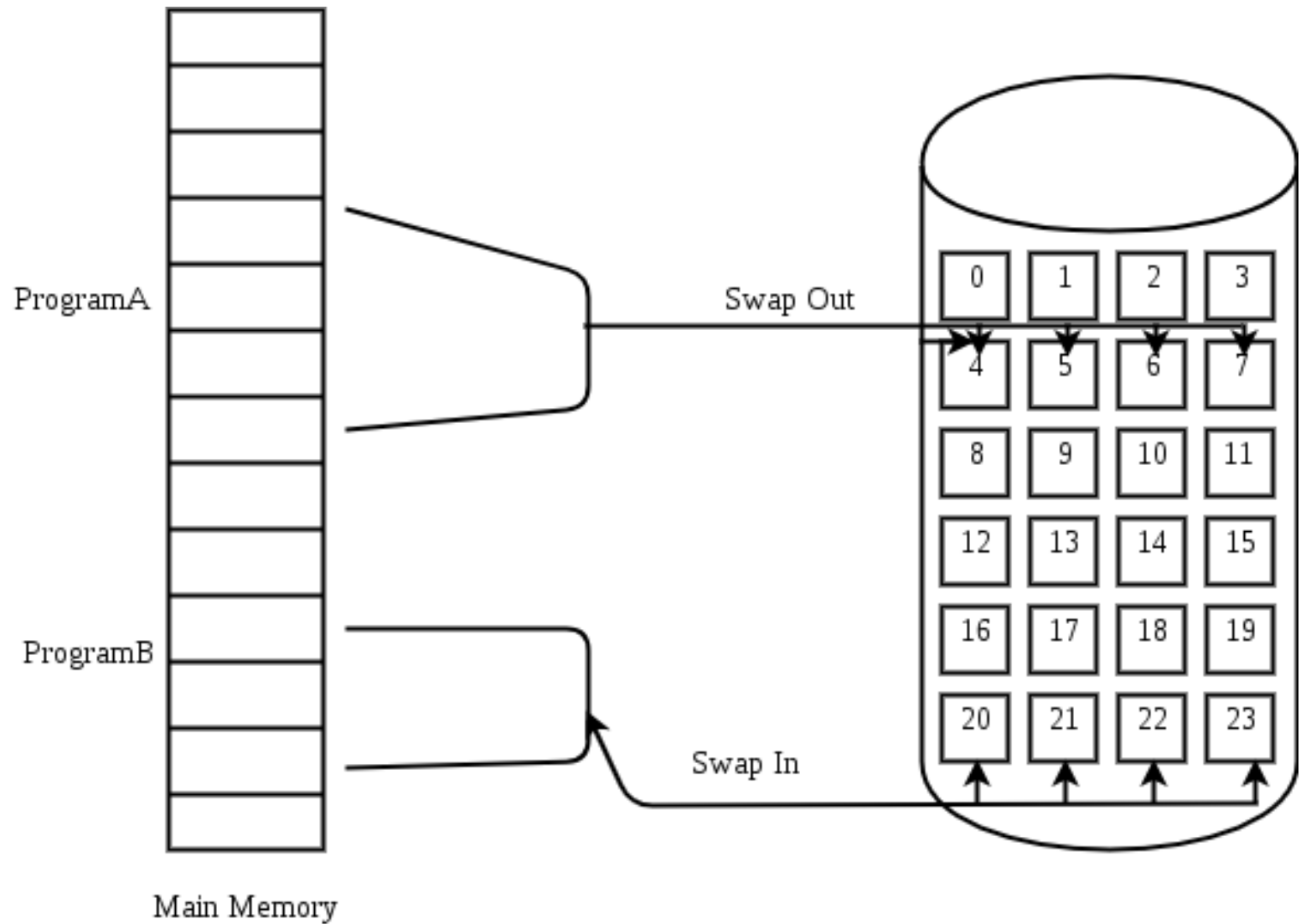
- Banyak proses yang dapat dikelola di main memory
- Sebuah proses boleh lebih besar dari main memory, dengan syarat pembatasan program harus dibuang

# Swapping

- Sebuah tindakan memindahkan sebuah proses keluar dari memori utama secara keseluruhan.
- Proses harus ditunda (suspend) untuk memastikan tidak berjalan ketika dipindahkan
- Ketika proses sibuk swap in/out maka kondisi ini disebut trashing



# Ilustrasi



# Penyebab Trashing

- Jumlah proses meningkat di memory melebihi frame yang sudah dialokasikan di tiap proses akan berkurang.
- Contoh
  - Frame bebas 400:
  - Jumlah proses 50
  - Frame per Proses : 8

# Algoritma Swapping

- FIFO
- LRU
- LFU
- MFU
- DLL

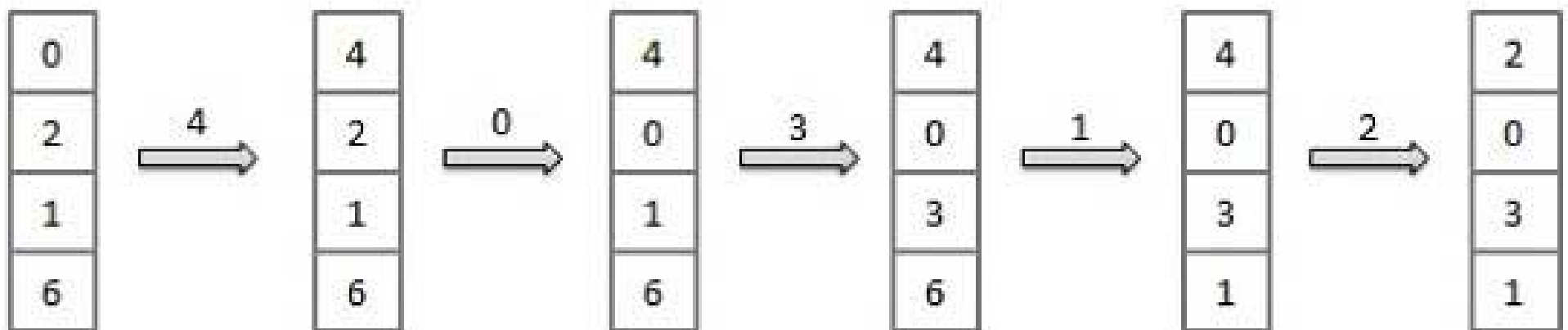
# First In First Out

- Page tertua di memori utama yang akan digantikan
- Mudah diimplementasikan, mengganti page dari ekor dan menambah dari kepala

# Ilustrasi

Reference String : 0, 2, 1, 6, 4, 0, 1, 0, 3, 1, 2, 1

Misses : x x x x x x x x x



$$\text{Fault Rate} = 9 / 12 = 0.75$$

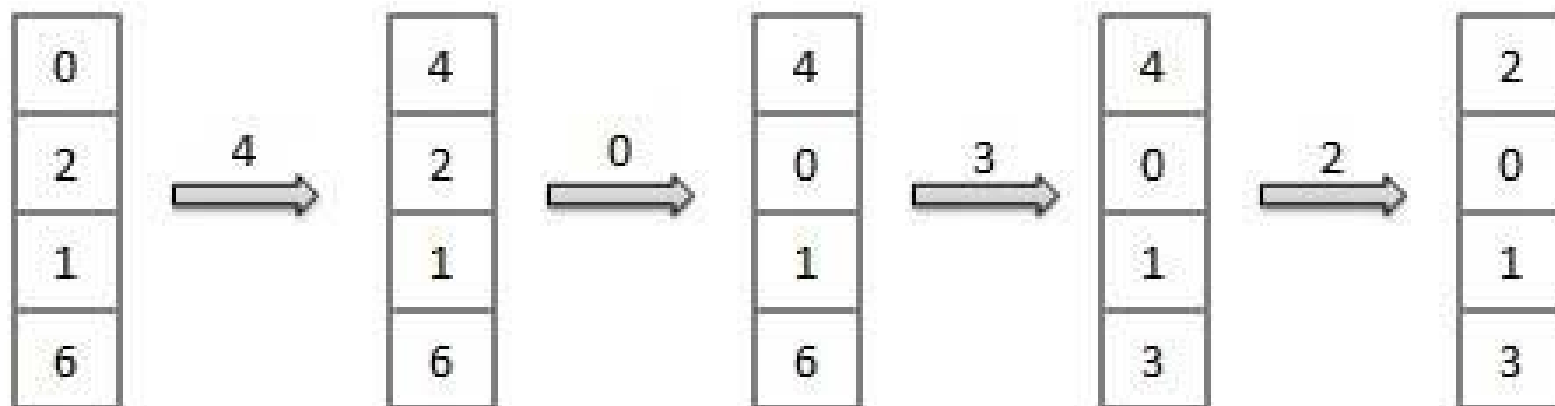
# Least Recently Used

- Page yang paling lama tidak digunakan di memori utama yang akan diganti
- Penggantian page berdasarkan waktu paling lama

# Ilustrasi

Reference String : 0, 2, 1, 6, 4, 0, 1, 0, 3, 1, 2, 1

Misses : x x x x x x x x



$$\text{Fault Rate} = 8 / 12 = 0.67$$

# Least Frequently Used

- Page dengan jumlah paling kecil yang akan digantikan
- Algoritma ini menderita dari situasi yang dimana sebuah halaman digunakan secara penuh selama inisialisasi proses, tetapi tidak pernah digunakan lagi



# Isu dari Implementasi

## Instruction backup

Instruksi yang menyebabkan referensi ke page yang belum ada di memori (menyebabkan page fault) harus diulang ketika page tersebut telah tersedia.

# Lanjutan

Locking pages in memory

Pada saat satu proses menjalani tahap I/O, proses lain bisa dijalankan.

# Lanjutan

## Shared pages

Dua atau lebih proses bisa memakai bersama page-page yang berasal dari editor yang mereka pakai.

# Lanjutan

## Backing Store

Pada disk, disediakan area untuk menampung page yang dikeluarkan dari memori (paged out) yang disebut swap area.

# Lanjutan

## Paging Daemon

Untuk meyakinkan tersedianya frame bebas yang cukup banyak, banyak sistem paging yang menggunakan proses background yang disebut paging daemon.