
Chapter 7

Operating System

(OS)

Definisi OS:

⌘ Suatu program yang mengatur eksekusi program-program aplikasi dan berfungsi sebagai interface antara pengguna komputer dengan hardware komputer

Tujuan dan fungsi

⌘ Kemudahan

- ☑ OS membuat komputer lebih mudah untuk digunakan

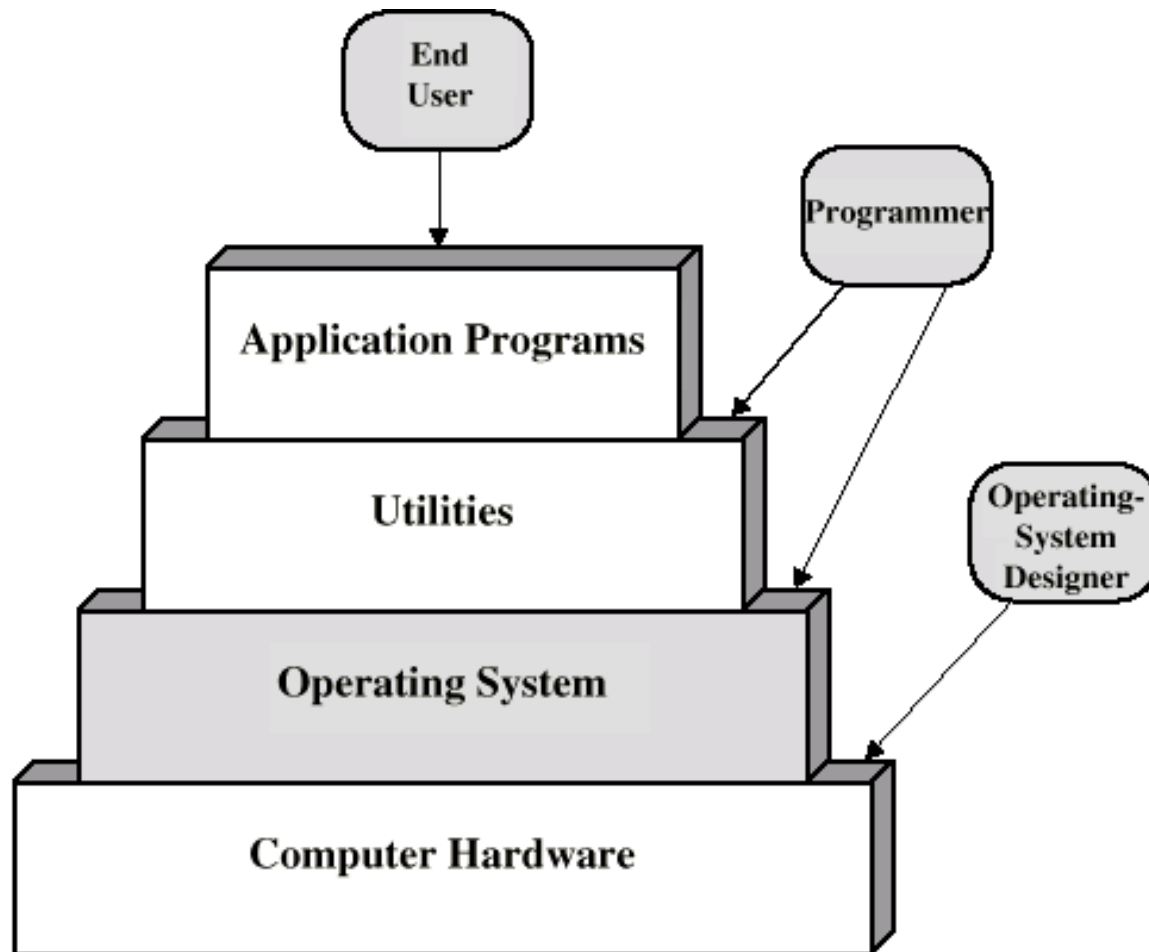
⌘ Efisiensi

- ☑ OS memungkinkan sumber daya sistem komputer digunakan dengan cara yang efisien

⌘ Kemampuan berkembang

- ☑ OS harus disusun sedemikian rupa shg memungkinkan pengembangan yang efektif, pengujian, dan penerapan fungsi sistem baru tanpa mengganggu layanan yang telah ada

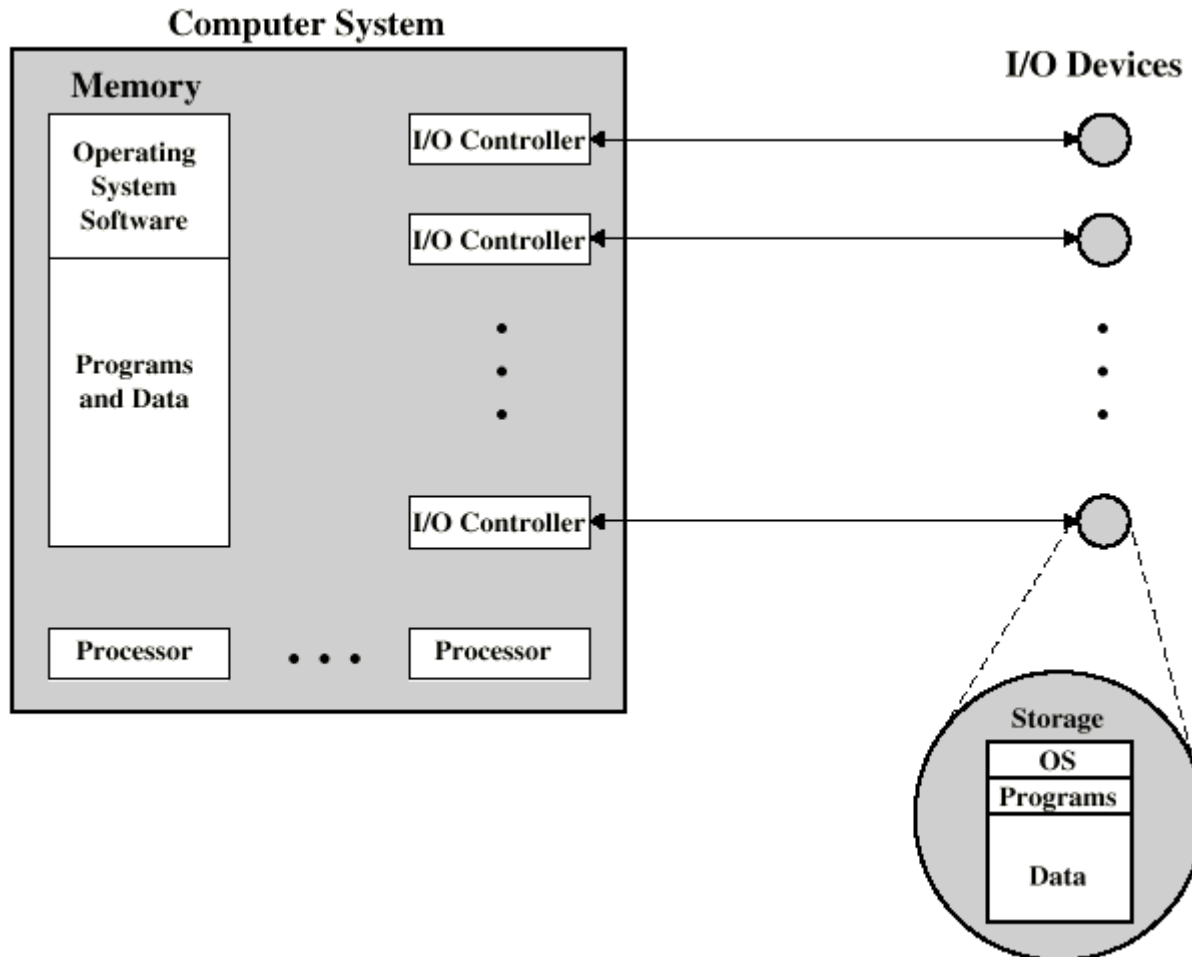
Layers dan Gambaran Sistem Komputer



Fungsi Sistem Operasi

- ⌘ Pembuatan program
- ⌘ Eksekusi program
- ⌘ Akses ke perangkat I/O
- ⌘ Akses terkontrol ke file
- ⌘ Akses sistem
- ⌘ Deteksi error dan respons
- ⌘ Laporan

O/S sebagai Pengatur Sumber Daya



Tipe Sistem Operasi

⌘ Interaktif

⌘ Batch

⌘ Single program (Uni-programming)

⌘ Multi-programming (Multi-tasking)

Sistem-sistem lama

- ⌘ akhir 1940 sampai pertengahan 1950
- ⌘ Tidak ada sistem operasi
- ⌘ Program berhubungan langsung dengan hardware
- ⌘ Terdapat dua masalah utama:
 - ☒ Penjadwalan (Scheduling)
 - ☒ Waktu Setup (Setup time)

Sistem Batch Sederhana

- ⌘ Program resident monitor
- ⌘ Pengguna mengajukan job ke operator
- ⌘ Operator mengumpulkan job
- ⌘ Monitor mengontrol rangkaian event untuk memproses kumpulan job
- ⌘ Setiap job dibuat bercabang agar kembali ke monitor apabila pengolahannya selesai, pada posisi ini monitor akan mulai memuatkan secara otomatis program berikutnya
- ⌘ Monitor menangani scheduling

Catatan:

- ⌘ Monitor : teknik sistem operasi batch dengan menggunakan potongan software
- ⌘ Job : sebuah program tunggal
- ⌘ Resident memory : bagian monitor yang berada di memori utama

Job Control Language (JCL)

⌘ Instruksi untuk Monitor

⌘ Biasanya diawali oleh \$

⌘ e.g.

⌘ \$JOB

⌘ \$FTN

⌘ ... Some Fortran instructions

⌘ \$LOAD

⌘ \$RUN

⌘ ... Some data

⌘ \$END

Fitur Hardware yang diperlukan

⌘ Proteksi Memori

- ☑ Untuk melindungi monitor

⌘ Timer

- ☑ Untuk mencegah job memonopoli sistem

⌘ Instruksi istimewa (Privileged instructions)

- ☑ Hanya dieksekusi Monitor

- ☑ e.g. I/O

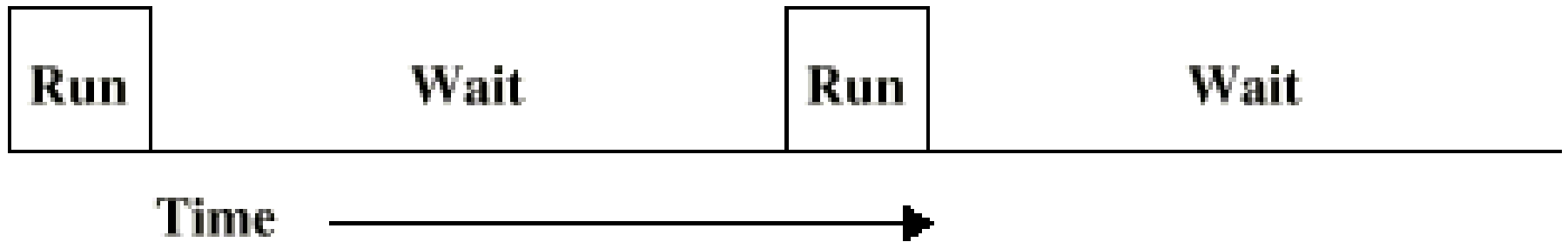
⌘ Interrupts

- ☑ Dibolehkan untuk melepaskan dan mendapatkan kembali kontrol

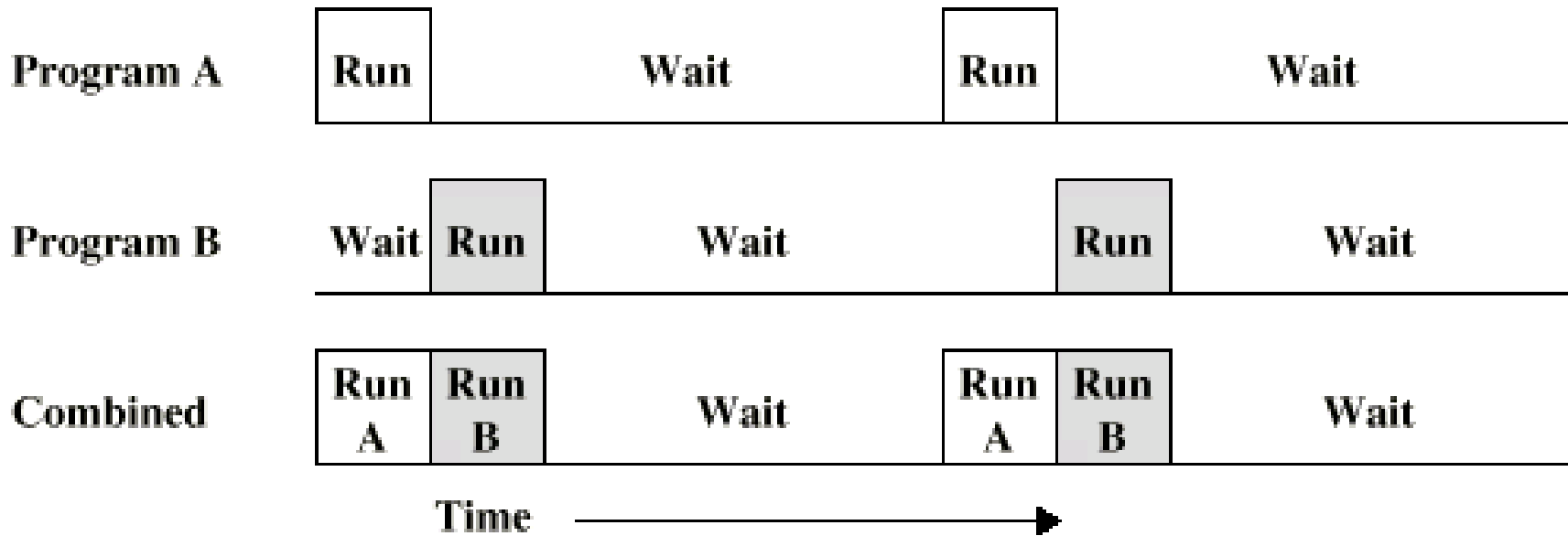
Sistem Batch Multi-programmed

- ⌘ Perangkat I/O sangat lambat
- ⌘ Ketika sebuah program menunggu untuk I/O, program lain bisa menggunakan CPU

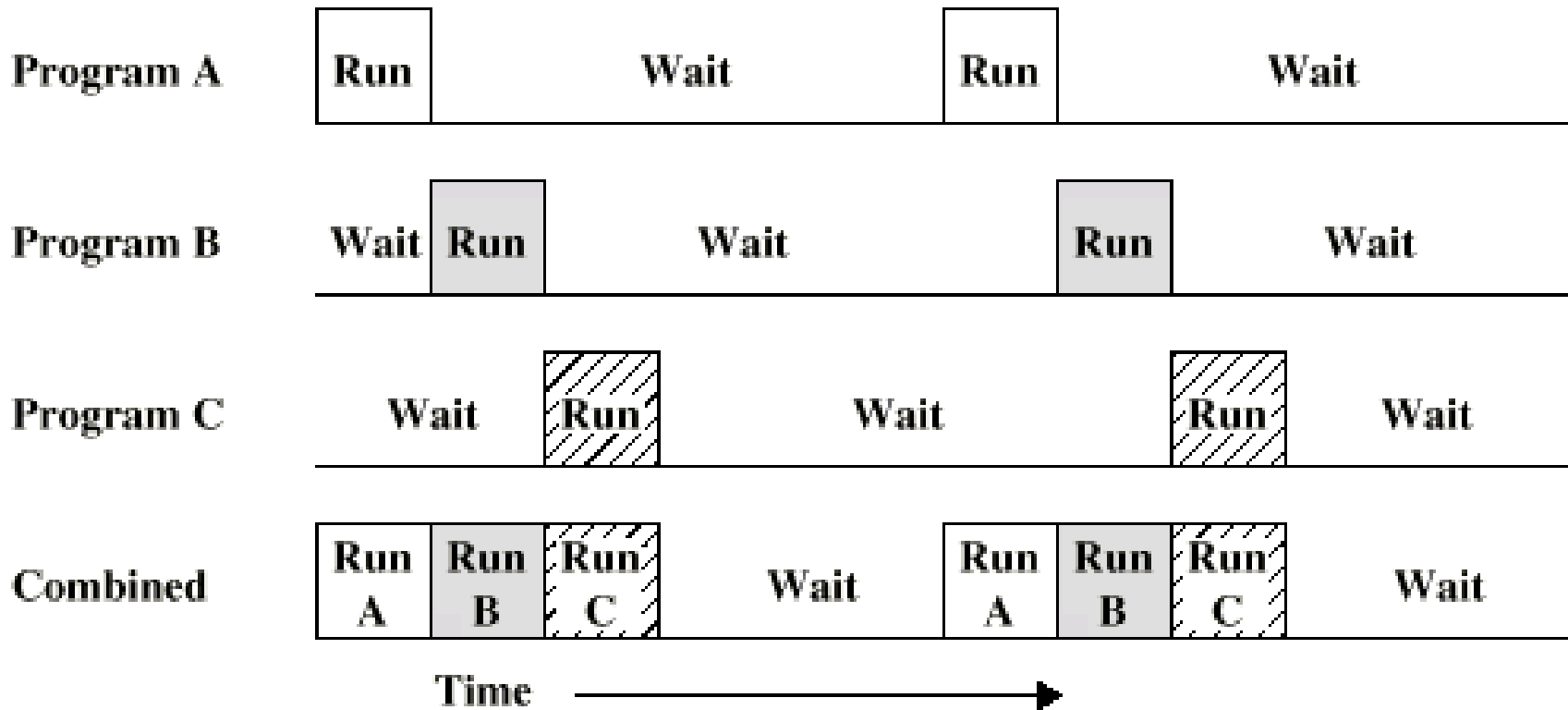
Single Program



Multi-Programming dengan Dua Program



Multi-Programming dengan Tiga Program



Sistem Pembagian Waktu

- ⌘ Mengizinkan pengguna untuk berkomunikasi langsung dengan komputer
 - ☒ i.e. Interactive
- ⌘ Multi-programming mengizinkan sejumlah pengguna untuk berkomunikasi dengan komputer

Penjadwalan (Scheduling)

- ⌘ Kunci untuk dapat multi-programming
- ⌘ Long term
- ⌘ Medium term
- ⌘ Short term
- ⌘ I/O

Penjadwalan Long Term

- ⌘ Ditetapkan dimana program diajukan untuk diproses
 - ☑ Derajat pengontrolan multi-programming
- ⌘ Saat diajukan, sebuah job akan diproses untuk pejadwalan short term
- ⌘ (atau akan menukar job untuk penjadwalan medium term)

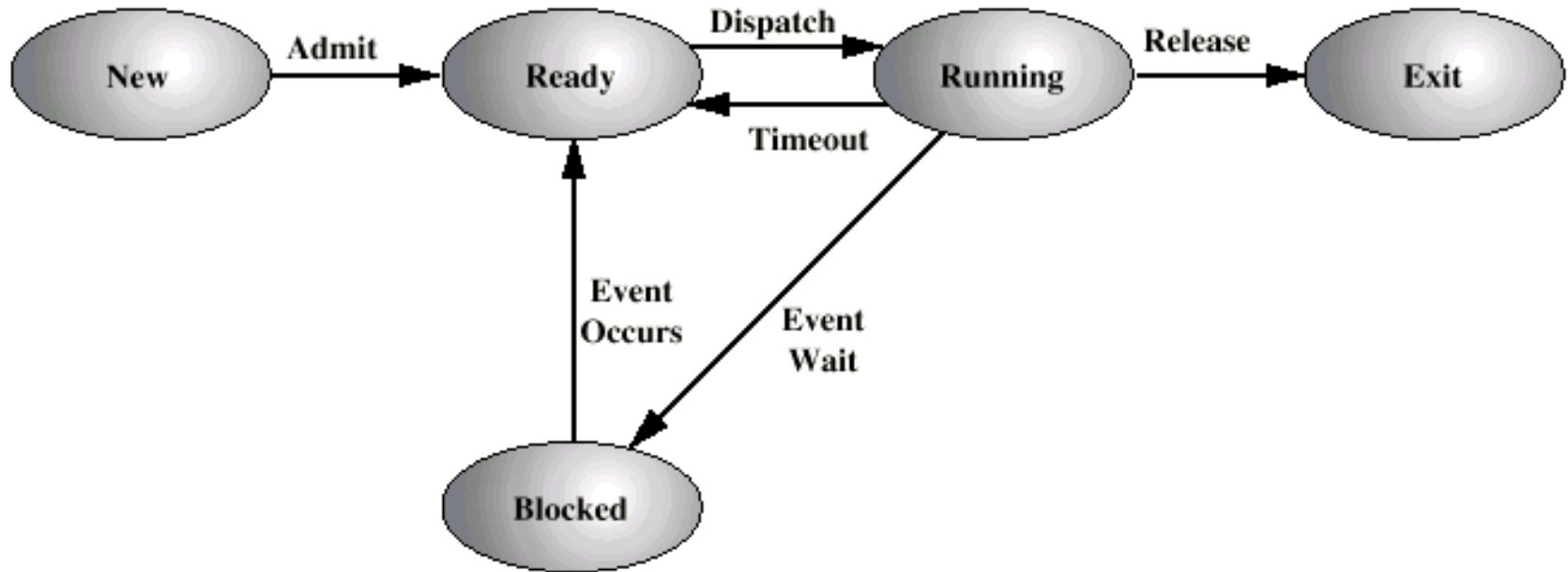
Penjadwalan Medium Term

- ⌘ Bagian dari fungsi penukaran
- ⌘ Biasanya berdasarkan pada kebutuhan untuk mengatur multi-programming
- ⌘ Jika tidak ada virtual memori, maka pengaturan memori juga sebuah isuu

Penjadwalan Short Term

- ⌘ Pengatur pengiriman berita (Dispatcher)
- ⌘ Fine grained decisions of which job to execute next
- ⌘ i.e. which job actually gets to use the processor in the next time slot

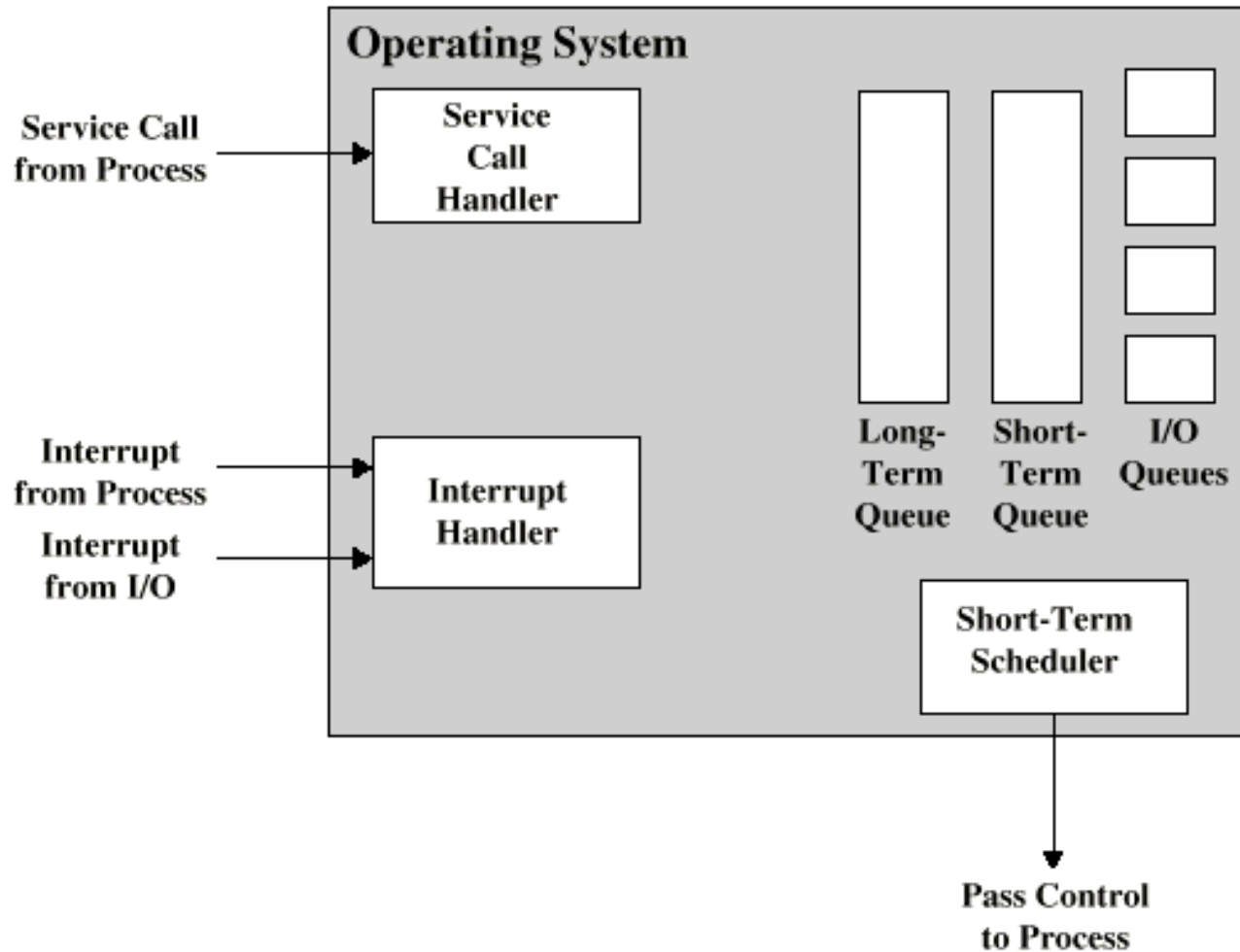
Gerbang Proses



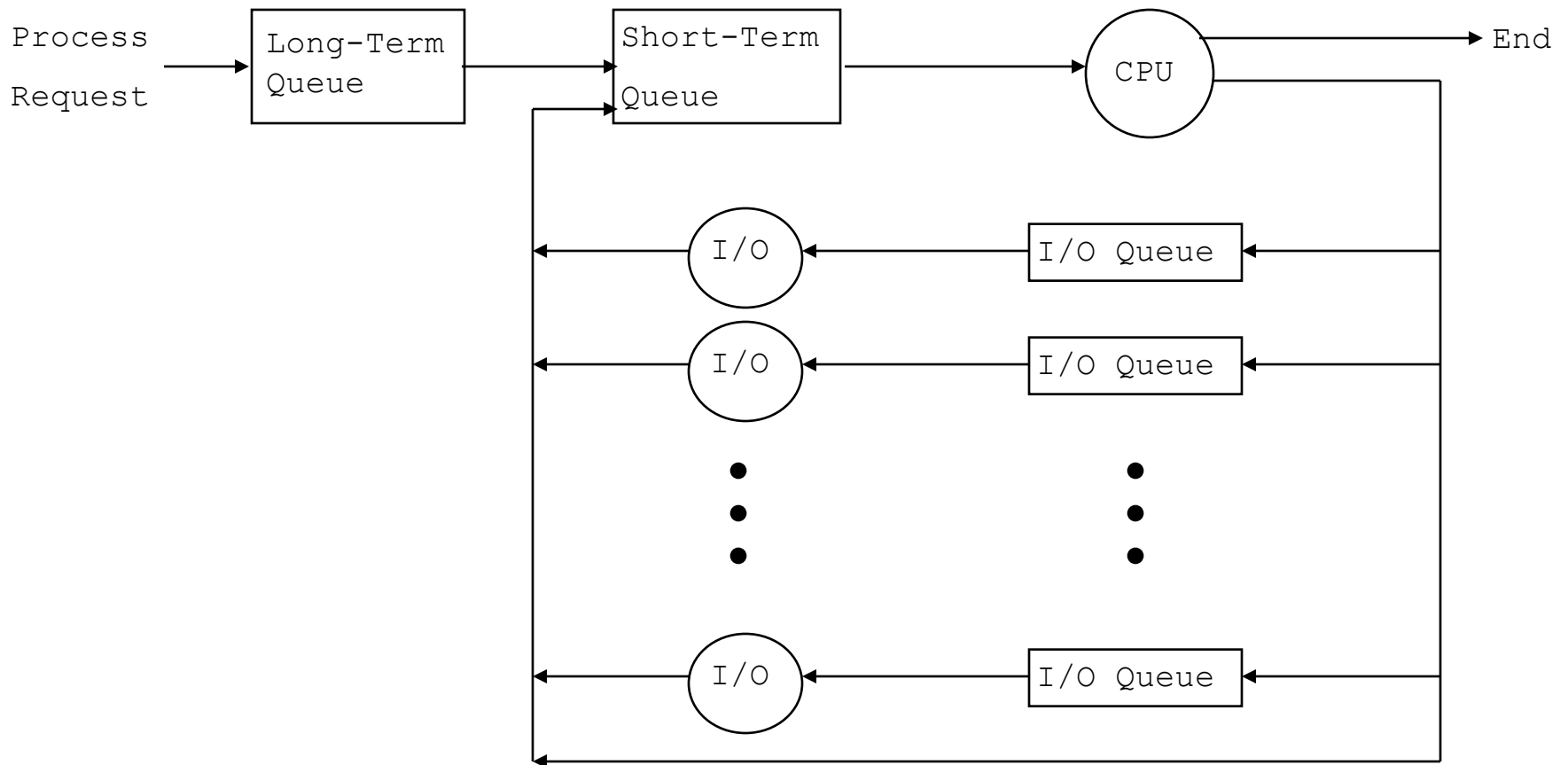
Process Control Block

- ⌘ Identifikatassi
- ⌘ Gerbang
- ⌘ Priori
- ⌘ Program counter
- ⌘ Memory pointers
- ⌘ Data Isi
- ⌘ Status I/O
- ⌘ Informasi laporan

Element-element O/S



Proses Penjadwalan



Managemen Memori

⌘ Uni-program (Program tunggal)

- ☑ Memori dipisah menjadi dua
- ☑ Satu untuk OS (monitor)
- ☑ Satu untuk eksekusi program

⌘ Multi-program

- ☑ “pengguna” merupakan bagian dari proses yang sedang aktif

Swapping (Penukaran)

⌘ Masalah: I/O lebih lambat dibandingkan CPU meskipun pada sistem multi-programming, CPU akan mempunyai banyak waktu kosong (tidak bekerja)

⌘ Solusi:

☑ Menambah memori utama

☒ Mahal

☒ Diutamakan untuk program besar

☑ Swapping

☒ memindahkan isi memori utama ke memori sekunder

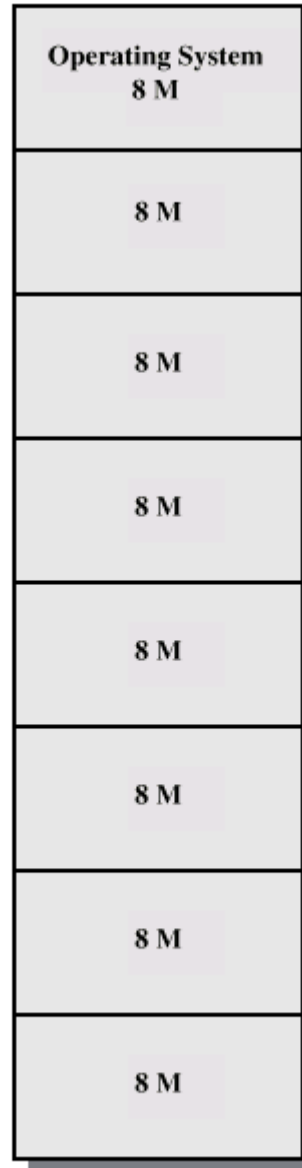
Apa itu Swapping?

- ⌘ Antrian Long term proses penyimpanan pada disk
- ⌘ Sebuah proses yang dikeluarkan sementara dari memori utama
- ⌘ Jika tidak ada proses pada memori utama dalam keadaan ready
(i.e. semua I/O di-blocked)
 - ☑ OS akan memindahkan proses yang di-block ke antrian, antrian ini adalah proses yang dikeluarkan sementara atau ditunda
 - ☑ Pemindahan pada proses *ready* atau sebuah proses baru
 - ☑ Tetapi swapping adalah proses I/O...

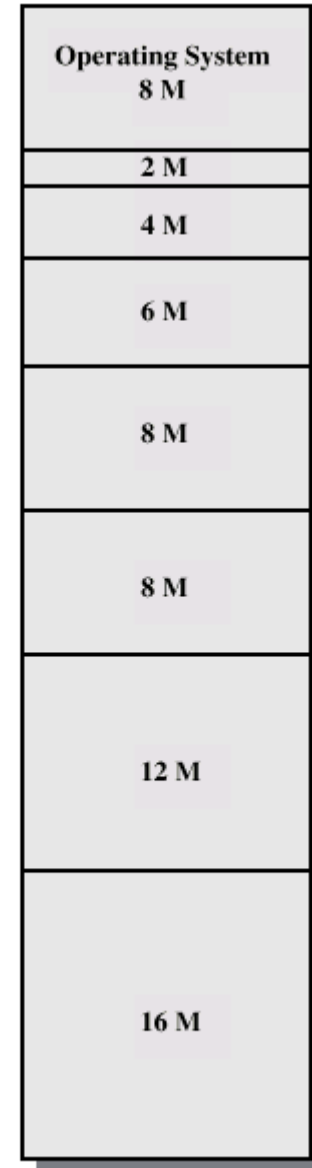
Pembagian (Partitioning)

- ⌘ Membagi memori kedalam bagian-bagian untuk alokasi pemrosesan (termasuk OS)
- ⌘ Fixed-sized partitions (Partisi tetap)
 - ☒ Membagi memori utama dengan ukuran yang tetap
 - ☒ Setiap proses yang berukuran kecil atau sama dapat menempati sembarang partisi
 - ☒ Jika partisi penuh, maka OS dapat men-swap beberapa proses dan memuatkan proses lain
 - ☒ Tidak efisien dalam penggunaan memori karena ada fragmentasi internal; jumlah proses aktif tetap

Fixed Partitioning



(a) Equal-size partitions



(b) Unequal-size partitions

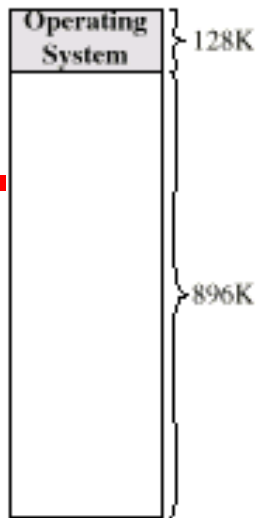
Partisi Dinamis (1)

- ⌘ Partisi yang digunakan memiliki panjang dan jumlah yang berbeda-beda
- ⌘ Proses dialokasikan ke dalam partisi yang ukurannya sama dengan proses
- ⌘ Terdapat ruang kosong di akhir memori, terlalu kecil untuk digunakan
 - ☒ Satu ruang kosong kecil – memori terbuang
- ⌘ Ketika semua proses di-bloked, OS akan men-swap sebuah proses dan menempatkan proses lain

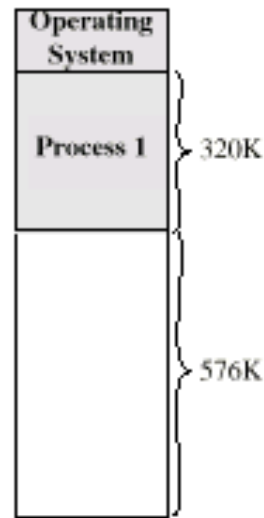
Variable Sized Partitions (2)

- ⌘ Proses baru ukurannya mungkin lebih kecil dari proses yang di-swap
- ⌘ Akan terjadi ruang kosong lagi
- ⌘ Akan menimbulkan banyak ruang kosong
- ⌘ Solusi:
 - ☑ Koalisi – menggabungkan semua ruang kosong menjadi ruang kosong yang besar
 - ☑ Kompaksi – OS menggeser proses-proses menjadi berada dalam satu blok, sehingga terdapat ruang kosong yang besar
(c.f. disk de-fragmentation)

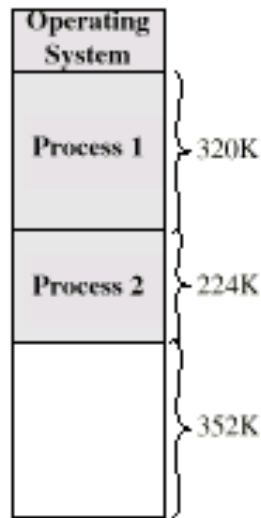
Effect of Dynamic Partitioning



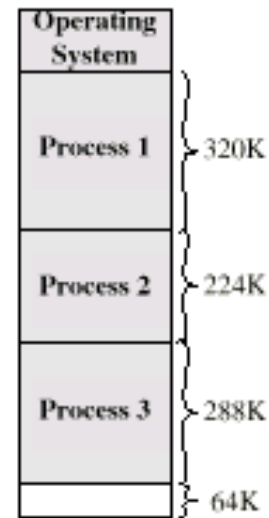
(a)



(b)



(c)



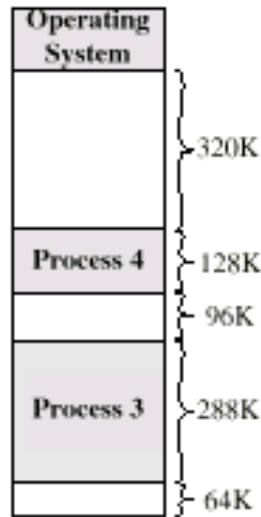
(d)



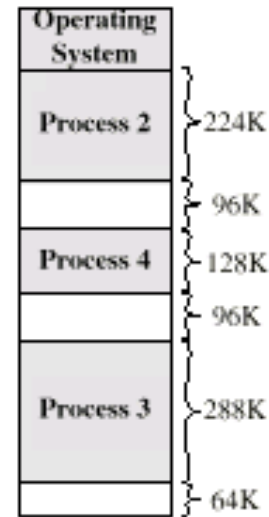
(e)



(f)



(g)



(h)

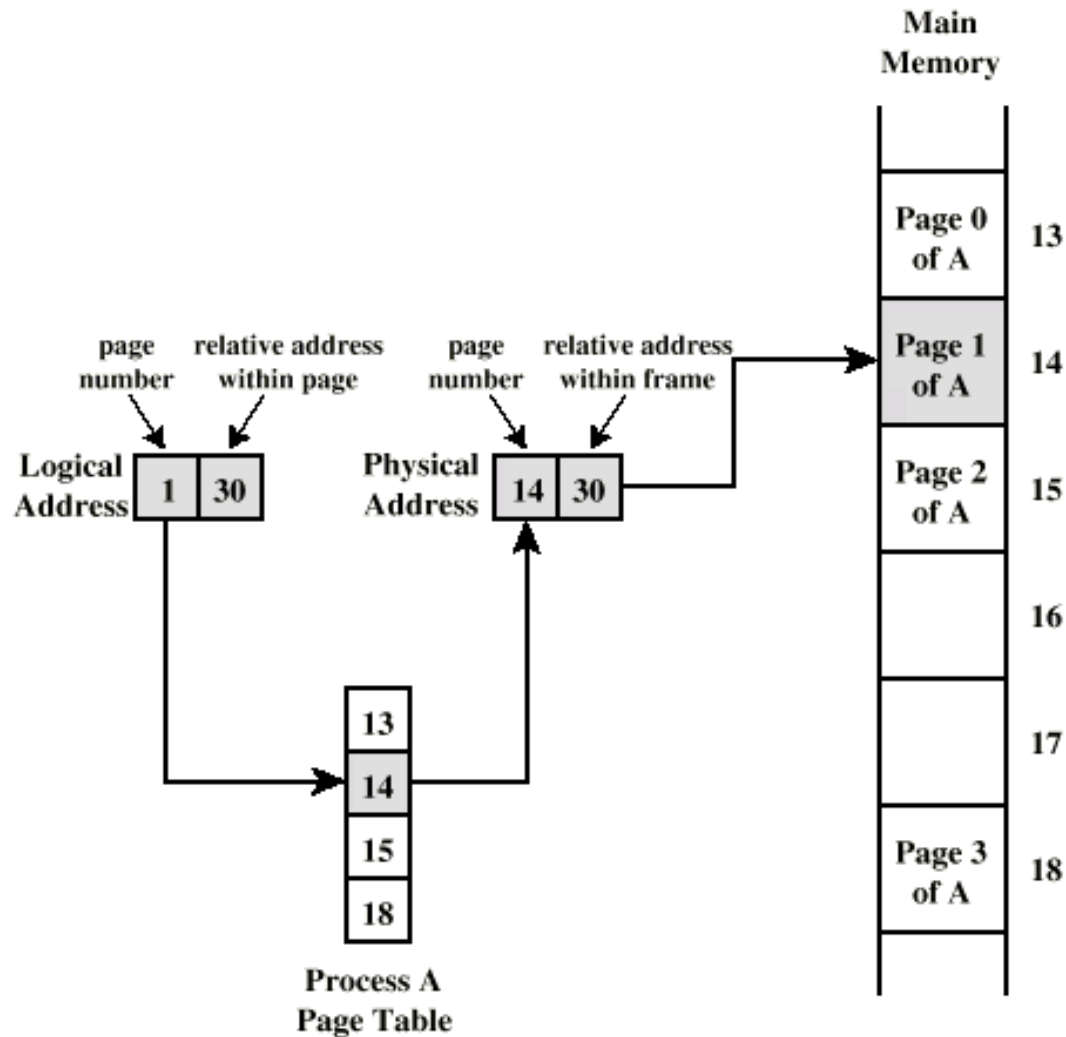
Relokasi

- ⌘ Tidak ada jaminan bahwa proses akan menempati partisi yang sama dalam memori
- ⌘ Instruksi terdiri dari alamat
 - ☑ Lokasi data
 - ☑ Alamat untuk instruksi (branching)
- ⌘ Alamat logika – referensi ke sebuah lokasi memori
- ⌘ Alamat fisik – lokasi aktual di dalam memori

Paging

- ⌘ Memori utama dibagi menjadi ukuran yang sama, chunk kecil - page frames
- ⌘ Program dibagi menjadi proses-proses berukuran sama (chunk) - pages
- ⌘ Mengalokasikan frame semua page ke sebuah proses
- ⌘ OS menjaga daftar frame yang bebas
- ⌘ Menggunakan page table untuk menunjukkan lokasi semua page proses

Logical and Physical Addresses - Paging



Virtual Memory

⌘ Tuntutan paging

- ☑ Tidak memuat semua page proses di dalam memori
- ☑ Hanya memuat page seperti yang diminta

⌘ Page fault

- ☑ Page yang diinginkan tidak berada dalam memori
- ☑ OS harus men-swap page yang diinginkan
- ☑ Diperlukan untuk men-swap sebuah page agar terdapat ruang kosong
- ☑ Menyeleksi page yang dikeluarkan berdasarkan history

Thrashing

- ⌘ Terlalu banyak proses pada memori yang kecil
- ⌘ OS terlalu banyak melakukan swapping
- ⌘ Kecil atau bahkan tidak ada proses yang selesai
- ⌘ Disk menyala sepanjang waktu

⌘ Solusi

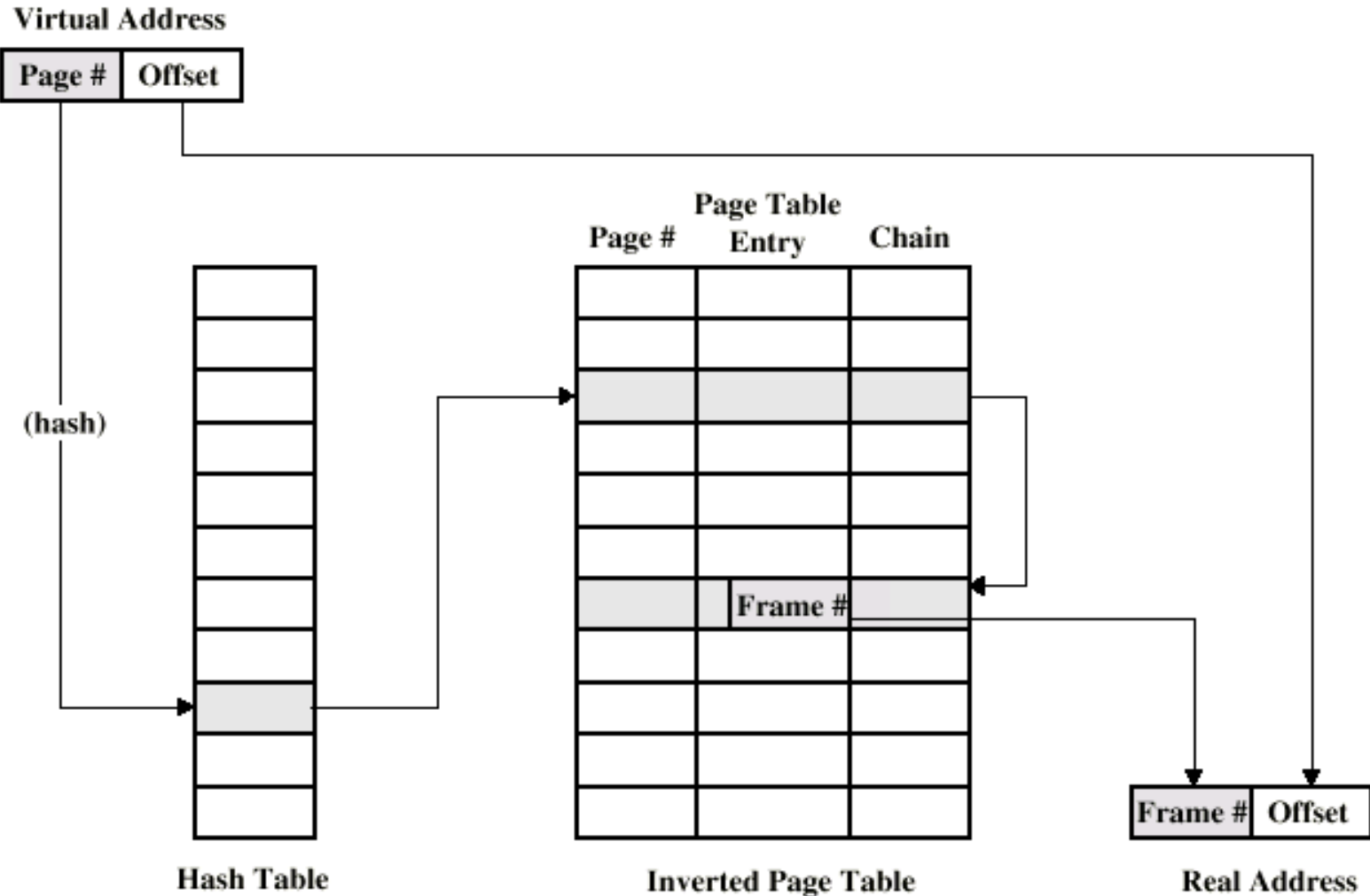
- ☑ Menggunakan algoritma penempatan yang baik
- ☑ Memperkecil jumlah proses yang bekerja

Catatan

- ⌘ Kita tidak memerlukan seluruh proses dalam memori untuk dijalankan
- ⌘ Kita dapat dalam pages seperti yang diinginkan
- ⌘ Maka kita dapat menjalankan proses lebih besar dari memori yang tersedia

- ⌘ Karena proses terjadi di memori utama, maka memori utama disebut real memory
- ⌘ Pengguna/programmer akan melihat memori yang besar - virtual memory

Struktur Page Table



Segmentasi

- ⌘ Paging adakalanya tidak berguna bagi programmer
- ⌘ Segmentasi dimungkinkan bagi programmer
- ⌘ Biasanya program dan data dialokasikan di segment yang berbeda
- ⌘ Segmentasi memungkinkan pemrogram menganggap memori terdiri dari kumpulan ruang alamat/segment

Keunggulan segmentasi

- ⌘ Menyederhanakan penanganan perkembangan struktur data
- ⌘ Memungkinkan program untuk diubah dan dikompilasi ulang secara independen, tanpa re-linking dan memuat kembali seluruh program
- ⌘ Membiarkan dirinya untuk berbagi pakai proses
- ⌘ Memungkinkan dilakukan proteksi
- ⌘ Beberapa sistem menggabungkan segmentasi dan paging

Required Reading

⌘ Stallings chapter 7

⌘ Stallings, W. Operating Systems, Internals and Design Principles, Prentice Hall 1998

⌘ Loads of Web sites on Operating Systems