

RANGKUMAN MATERI
PROSES - PROSES SISTEM OPERASI



oleh :

Khosyi Nasywa Imanda

2131710103

MI-1E

PROGRAM STUDI D-III MANAJEMEN INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG

Jl. Soekarno Hatta No.9, Jatimulyo, Kec, Lowokwaru, Kota Malang, Jawa Timur

65141

A. KONSEP PROSES

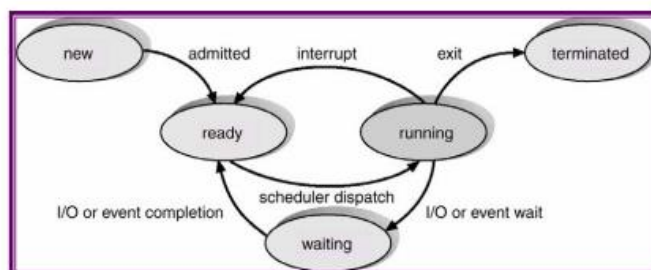
- Proses merupakan semua aktivitas CPU meliputi:
 - ✓ Jobs yang dieksekusi pada sistem batch
 - ✓ User program atau task pada sistem time-shared
 - ✓ Beberapa program yang dijalankan pada satu waktu: satu program interaktif dan beberapa program batch pada sistem single user seperti MS-DOS dan Macintosh O.S
- Sistem operasi mengeksekusi berbagai jenis program. Pada sistem batch program tersebut biasanya disebut dengan job, sedangkan pada sistem time sharing, program disebut dengan program user atau task. Beberapa buku teks menggunakan istilah job atau proses.

B. PROSES

- Proses adalah program yang sedang dieksekusi oleh processor , Eksekusi proses dilakukan secara berurutan sehingga satu instruksi dieksekusi sesuai kepentingan proses
- **Eksekusi proses** dilakukan secara berurutan **berdasarkan prioritas proses**.
- Sumber daya yang dibutuhkan seperti: CPU time, memori, file, dan I/O
- Sebuah program **tidak dapat** disebut sebuah proses, **karena** program hanyalah entity pasif seperti file yang disimpan pada disk. **Sedangkan** proses adalah entity aktif yang mempunyai program counter yang menunjuk ke instruksi selanjutnya yang akan dieksekusi dan memiliki sekumpulan resource

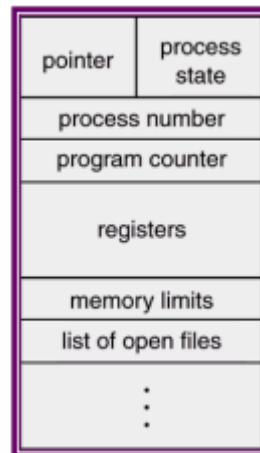
C. STATUS PROSES (*Process State*)

- Meskipun tiap-tiap proses terdiri dari suatu kesatuan yang terpisah namun adakalanya proses-proses tersebut butuh untuk saling berinteraksi. Satu proses bisa dibangkitkan dari output proses lainnya sebagai input.
- Pada saat proses dieksekusi, akan terjadi perubahan status. Status proses didefinisikan sebagai bagian dari aktivitas proses yang sedang berlangsung saat itu. Gambar 3-1 menunjukkan diagram status proses. Status proses terdiri dari :
 - ✓ New: proses sedang dibuat.
 - ✓ Running: proses sedang dieksekusi.
 - ✓ Waiting: proses sedang menunggu beberapa event yang akan terjadi (seperti menunggu untuk menyelesaikan I/O atau menerima sinyal).
 - ✓ Ready: proses menunggu jatah waktu dari CPU untuk diproses.
 - ✓ Terminated: proses telah selesai dieksekusi.



D. PCB (*Process Control Block*)

- Masing-masing proses direpresentasikan oleh Sistem Operasi dengan menggunakan Process Control Block (PCB), seperti yang terlihat pada Gambar berikut ini.



- ✓ Status Proses. New, ready, running, waiting dan terminated.
- ✓ Program Counter. Menunjukkan alamat berikutnya yang akan dieksekusi oleh proses tersebut.
- ✓ CPU Registers. Register bervariasi tipe dan jumlahnya tergantung arsitektur komputer yang bersangkutan. Register-register tersebut terdiri atas: accumulator, index register, stack pointer, dan register serbaguna dan beberapa informasi tentang kode kondisi.
- ✓ Informasi Penjadwalan CPU. Informasi tersebut berisi prioritas dari suatu proses, pointer ke antrian penjadwalan, dan beberapa parameter penjadwalan yang lainnya.
- ✓ Informasi Manajemen Memori. Informasi tersebut berisi nilai (basis) dan limit register, page table, atau segment table tergantung pada sistem memory yang digunakan oleh SO.
- ✓ Informasi Accounting. Informasi tersebut berisi jumlah CPU dan real time yang digunakan, time limits, account numbers, jumlah job atau proses, dll.

E. PENJADWALAN PROSES

- Penjadwalan proses merupakan pembagian waktu agar suatu proses dapat berjalan sepanjang waktu dan penggunaan CPU menjadi lebih maksimal. Tujuan pembagian waktu adalah untuk mengganti CPU di antara proses-proses yang begitu sering dilakukan, sehingga pengguna dapat berinteraksi dengan program lain saat CPU sedang bekerja.



- ✓ Scheduling Queue , Secara umum, antrian penjadwalan diklasifikasikan dalam 3 kategori:

- ✚ **Job Queue:** antrian berisi semua proses yang memasuki sistem
- ✚ **Ready Queue:** tempat proses-proses di memori utama yang menunggu dieksekusi
- ✚ **Device Queue:** deretan proses yang sedang menunggu peralatan I/O tertentu

Setiap antrian disimpan sebagai linked list. Header berisi pointer PCB pertama dan terakhir dari list. Setiap PCB mempunyai pointer yang menunjuk ke proses berikutnya pada antrian.

- ✓ Scheduler , Scheduler dilibatkan untuk memilih proses dari scheduling queue. Secara umum, terdapat 3 tipe scheduler:
 - ✚ Long-term Schedulers (job scheduler)
 - ✚ Short-term Schedulers (CPU scheduler)
 - ✚ Medium-term Schedulers
- ✓ Context Switch . Context switch berisi perubahan pointer ke register set saat ini. Jika terdapat proses aktif yang lebih banyak dibandingkan register set, sistem meng-copy data register ke dan dari memori

F. OPERASI PADA PROSES

- OS menyediakan mekanisme untuk membuat (create) dan menghentikan (terminate) proses, karena proses-proses pada sistem dapat dieksekusi secara konkuren dan harus dibuat dan dihapus secara dinamis.



- ✓ Create Process (Pembuatan Proses) .
 Selama eksekusi, proses kemungkinan membuat beberapa proses baru menggunakan system call create-process .
 Setiap proses baru membuat proses lain membentuk “tree” dari proses.
 Beberapa event yang terjadi ketika proses membuat sub-proses:
 - ✚ Sub-proses dapat memperoleh resource langsung dari OS, atau terbatas pada subset dari resource pada proses parent
 - ✚ Parent harus membagi resource (seperti memory) di antara child-nya
 - ✚ Resource fisik dan logika yang berbeda yang diperoleh saat dibuat, inialisasi data (input) dilewatkan oleh proses parent ke proses child
- ✓ Terminate Process (Penghentian Proses)
 Proses dihentikan apabila selesai mengeksekusi pernyataan terakhir dan meminta OS untuk menghapusnya menggunakan system call “exit”.
 Parent menghentikan eksekusi satu dari child untuk sejumlah alasan, misalnya:
 - ✚ Child melebihi penggunaan dari beberapa resource yang sudah dialokasikan.
 - ✚ Task diserahkan ke child yang tidak lagi diperlukan

- ✚ Parent tersedia, OS tidak mengizinkan child melanjutkan jika parent diterminasi

✓ Cooperating Process

Proses-proses yang kongruen yang dieksekusi di OS berupa proses-proses yang terpisah (independence) atau saling bekerja sama (cooperate)

- ✚ **Proses terpisah (independent)** : jika tidak dapat berakibat atau diakibatkan oleh proses lain yang dieksekusi di sistem. Berarti proses yang tidak membagi semua data dengan proses lain.

- ✚ **Proses bekerjasama (cooperate)** : jika dapat berakibat atau diakibatkan oleh proses lain yang dieksekusi di sistem. Berarti proses yang membagi data dengan proses lain.

Lingkungan yang memungkinkan proses bekerjasama dilatarbelakangi beberapa alasan sbb :

- ✚ **Sharing informasi** : karena beberapa user tertarik pada sebagian informasi (seperti file sharing), harus disediakan lingkungan untuk mengizinkan akses konkuren untuk tipe resource ini
- ✚ **Kecepatan komputasi** : jika kita ingin menjalankan task tertentu lebih cepat, kita harus membagi menjadi sejumlah sub task, yang dieksekusi paralel. Kecepatan dapat dicapai jika komputer mempunyai elemen multiple processing
- ✚ **Modularity** : jika ingin membangun sistem modular, fungsi sistem dibagi ke beberapa proses terpisah
- ✚ **Kenyamanan** : user individual mungkin mempunyai beberapa task yang bekerja pada satu waktu, contohnya user edit, print, compile secara paralel

✓ Komunikasi Antar Proses

Sistem operasi menyediakan layanan proses untuk berkomunikasi dengan proses lain melalui Interprocess communication (IPC).

✚ **Struktur Dasar**

- Fungsi message system memungkinkan komunikasi antar proses tanpa perlu tempat untuk sharing variable
- Saluran komunikasi harus terdapat antara 2 proses yang akan berkomunikasi. Saluran dapat diimplementasikan dalam beberapa cara yaitu komunikasi langsung dan tak langsung.

✚ **Komunikasi Langsung (Direct Communication)**

- Setiap proses yang ingin berkomunikasi harus mencantumkan nama penerima atau pengirim secara eksplisit.
- Primitif send dan receive didefinisikan dengan (symmetric addressing) :
 - $\text{send}(P, \text{pesan})$: mengirim pesan ke proses P
 - $\text{receive}(Q, \text{pesan})$: menerima pesan dari proses Q
- Contoh : Producer-consumer problem

✚ **Komunikasi Tak Langsung (Indirect Communication)**

- Message dikirim atau diterima dari mailbox (disebut juga port)

- Mailbox dapat dipandang sebagai obyek dimana pesan dapat ditempatkan oleh proses dan pesan dihapus oleh proses
- Setiap mailbox mempunyai identifikasi yang unik yaitu primitif send dan receive didefinisikan sbb :
 - $\text{send}(A, \text{pesan})$: mengirim pesan ke mailbox A
 - $\text{receive}(A, \text{pesan})$: menerima pesan dari mailbox A

✓ Buffering

Saluran (link) mempunyai kapasitas yang menentukan jumlah pesan yang dapat ditampung sementara. Properti ini dapat dipandang sebagai antrian dari pesan dalam bentuk link diimplementasikan dalam 3 cara :

Zero capacity (no buffering)

Antrian mempunyai maksimum panjang 0 sehingga link tidak mempunyai pesan yang menunggu.

Bounded capacity (automatic buffering)

Antrian memiliki panjang maksimum tertentu (n)

Unbounded capacity (automatic buffering)

Antrian memiliki panjang tidak terbatas, sehingga sembarang pesan dapat menunggu di link dan pengiriman tidak pernah menunda pekerjaan.

✓ Kondisi Pengecualian

Sistem pesan sangat bermanfaat pada lingkungan terdistribusi, dimana proses berada pada site (mesin) yang berbeda.

Jika terjadi kegagalan baik pada sistem sentralisasi maupun terdistribusi, beberapa recovery error (exception-condition handling) harus ada :

Terminasi proses

Baik pengirim maupun penerima bisa menghentikan sebelum sebuah pesan diproses.

Situasi ini akan meninggalkan pesan yang tidak pernah diterima atau proses menunggu pesan yang tidak pernah dikirim

Kehilangan pesan

Pesan dari proses P ke proses Q bisa hilang di jaringan komunikasi, karena hardware atau kabel komunikasi gagal

Pesan Teracak

Pada saat pesan dikirim ke tujuan, tetapi diacak ditengah jalan (sebagai contoh, karena noise pada channel komunikasi)

Hal ini sama kasusnya dengan kehilangan pesan, OS akan mentransmisikan kembali pesan original