

A[Download PDF](#)[Full PDF Package](#)[Translate](#)

MAKALAH SISTEM OPERASI PROCESS AND MANAGEMENT PROCESS Diusulkan oleh: YOGA ARADA 19312252 PROGRAM STUDI S1 INFORMATIKA FAKULTAS TEKNIK DAN ILMU KOMPUTER UNIVERSITAS TEKNOKRAT INDONESIA

By Yoga Arada and Wulandia Lestari

48 Views 12 Pages 1 File ▾

2020, Yoga

[Download PDF](#)[Full PDF Package](#)[Translate](#)

⋮

Original PDF

Related

**MAKALAH
SISTEM OPERASI
PROCESS AND MANAGEMENT PROCESS**

Diusulkan oleh:
YOGA ARADA
19312252



**PROGRAM STUDI S1 INFORMATIKA
FAKULTAS TEKNIK DAN ILMU KOMPUTER
UNIVERSITAS TEKNOKRAT INDONESIA
TAHUN AJARAN 2020-2020**

Alhamdulillahirabbil'alamin, Puji syukur kehadirat Allah SWT, atas segala rahmat dan hidayah-Nya. Shalawat serta salam senantiasa tercurahkan kepada junjungan Nabi Agung Muhammad SAW yang selalu kita nantikan syafa'atnya di akhirat nanti.

Penulis mengucapkan syukur kepada Allah SWT atas limpahan nikmat sehat-Nya, baik itu berupa sehat fisik maupun akal pikiran, sehingga penulis mampu untuk menyelesaikan pembuatan makalah sebagai tugas Quiz dari mata kuliah Sistem Oprasi “PROCESS AND MANAGEMENT PROCESS”.

Penulis tentu menyadari bahwa makalah ini masih jauh dari kata sempurna dan masih banyak terdapat kesalahan serta kekurangan di dalamnya. Untuk itu, penulis mengharapkan kritik serta saran dari pembaca untuk makalah ini, supaya makalah ini nantinya dapat menjadi makalah yang lebih baik lagi. Demikian, dan apabila terdapat banyak kesalahan pada makalah ini penulis mohon maaf yang sebesar-besarnya.

Wassalamu'alaikum Wr. Wb.

HALAMAN JUDUL**KATA PENGANTAR****DAFTARN ISI****PENGERTIAN PROSES****STATUS PROSES**

Proses Control Block

PENJADWALAN PROSES

Antrian Penjadwalan
Penjadwalan (Scheduler)
Contex switch

PEMBUATAN PROSES

Pempuatan Proses
Penghentian Proses

Proses pada Sistem Operasi berisikan instruksi, program counter, data, register pemroses, alamat pengiriman, stack data dan variabel pendukung lainnya. Berikut ini beberapa pengertian tentang proses, diantaranya adalah :

- A. Proses adalah sebuah konsep utama pada sistem operasi, sehingga masalah manajemen proses adalah masalah utama pada perancangan sistem operasi.
- B. Proses adalah program yang sedang dieksekusi.
- C. Proses adalah unit kerja terkecil yang secara individu mempunyai sumber daya dan dijadwalkan oleh sistem operasi. Untuk kegiatan proses, sistem operasi berperan untuk mengelola segala proses pada sistem dan mengalokasikan sumber daya ke proses tersebut. Berbagai proses berjalan secara bersamaan, dimana setiap proses mendapatkan bagian-bagian memori dan kendali sendiri-sendiri (peran SO), sehingga setiap proses (program) memiliki 2 prinsip di bawah ini :
 - a) Independent, adalah program-program tersebut berdiri sendiri, tidak saling bergantung dan terpisah.
 - b) One program at any instant, adalah hanya satu proses yang dilayani oleh pemroses pada satu waktu.
 - c) Pada multiprogramming, teknik penanganan proses adalah dengan menjalankan satu proses dan secara cepat berpindah ke proses lainnya (bergiliran), sehingga menimbulkan efek paralel semu (pseudoparallelism).

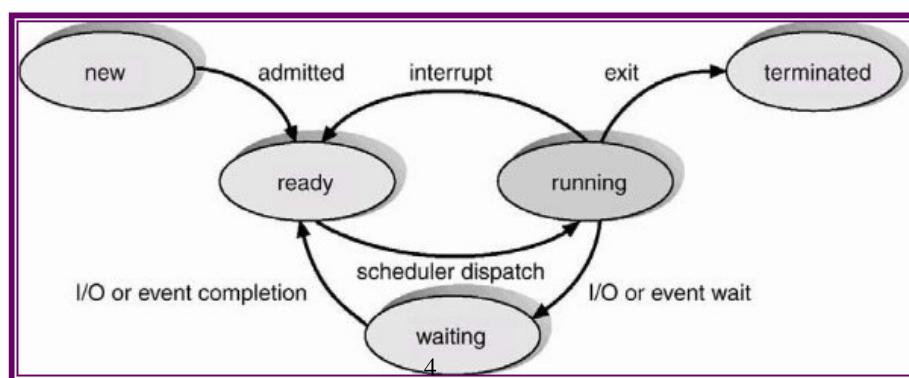
Meskipun pada setiap proses terdiri dari suatu kesatuan yang terpisah, tetapi adakalanya proses-proses tersebut butuh saling berinteraksi. Satu proses dapat dibangkitkan dari output proses lainnya sebagai input.

2. STATUS PROSES

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.

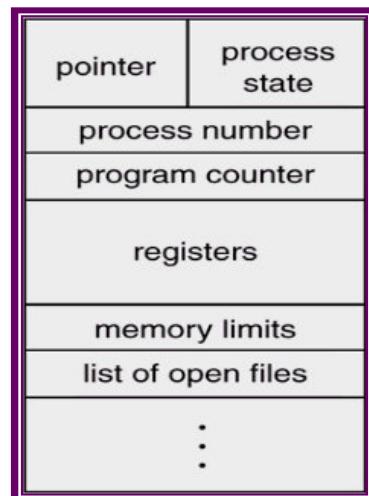
Pada saat proses dieksekusi, akan terjadi perubahan status. Status proses didefinisikan sebagai bagian dari aktivitas proses yang sedang berlangsung saat itu. Gambar 1 menunjukkan diagram status proses. Status proses terdiri dari :

- a. New: proses sedang dibuat.
- b. Running: proses sedang dieksekusi.
- c. Waiting: proses sedang menunggu beberapa *event* yang akan terjadi (seperti menunggu untuk menyelesaikan I/O atau menerima sinyal).
- d. Ready: proses menunggu jatah waktu dari CPU untuk diproses.
- e. Terminated: proses telah selesai dieksekusi.



Process Control Block (PCB)

Masing-masing proses direpresentasikan oleh Sistem Operasi dengan menggunakan *Process Control Block* (PCB), seperti yang terlihat pada Gambar 2

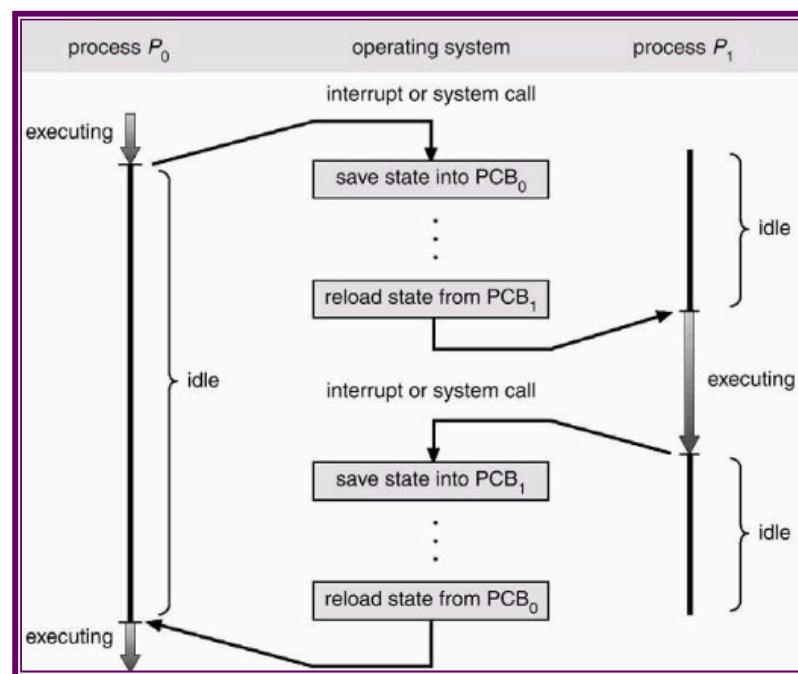


Gambar 2: Process Control Block

Informasi yang terdapat pada setiap proses meliputi :

- a. Status Proses. *New, ready, running, waiting* dan *terminated*.
- b. Program Counter. Menunjukkan alamat berikutnya yang akan dieksekusi oleh proses tersebut.
- c. 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.
Selama *Program Counter* berjalan, status informasi harus disimpan pada saat terjadi interrupt. Gambar 3 menunjukkan *switching* proses dari satu proses ke proses berikutnya.
- d. Informasi Penjadwalan CPU. Informasi tersebut berisi prioritas dari suatu

- e. Informasi Manajemen Memori. Informasi tersebut berisi nilai (basis) dan limit register, page table, atau segment table tergantung pada sistem memory yang digunakan oleh SO.
- f. Informasi Accounting. Informasi tersebut berisi jumlah CPU dan real time yang digunakan, time limits, account numbers, jumlah job atau proses, dll.
- g. Informasi Status I/O. Informasi tersebut berisi deretan I/O device (seperti tape driver) yang dialokasikan untuk proses tersebut, deretan file yang dibuka, dll.

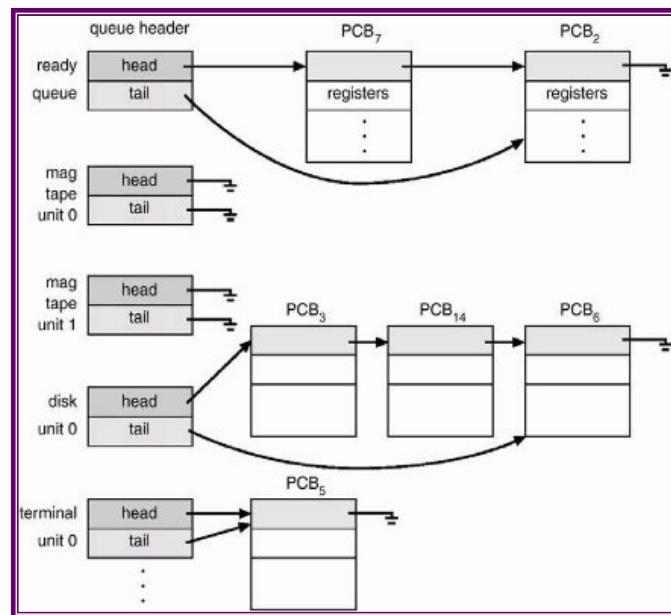


Gambar 3: Perpindahan CPU dari satu proses ke proses lain.

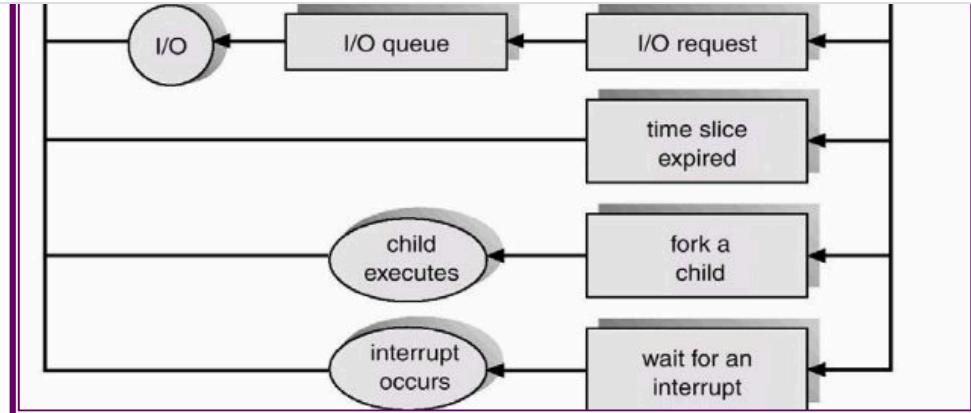
Penjadwalan direpresentasikan dalam bentuk antrian yang disimpan sebagai linkedlist dan berisi pointer awal dan akhir PCB. Tiap-tiap PCB memiliki suatu pointer field yang menunjuk ke proses berikutnya. Jenis-jenis antrian penjadwalan adalah sebagai berikut :

- *Job Queue*. Semua proses yang masuk pada suatu sistem akan diletakkan ke dalam *job queue*.
- *Ready Queue*. Sedangkan proses-proses yang ada di memori utama dan menunggu untuk dieksekusi diletakkan pada suatu list yang disebut dengan *ready queue*. Pada antrian ini berisi
- *Device Queue*. Deretan proses yang sedang menunggu peralatan I/O tertentu disebut dengan *device queue*.

Setiap proses dapat berpindah dari satu antrian ke antrian lain. Gambar 4 menunjukkan contoh *ready queue* dan *device queue*. Representasi dari penjadwalan proses dapat dilihat pada Gambar 5



Gambar 4: Ready queue dan device queue



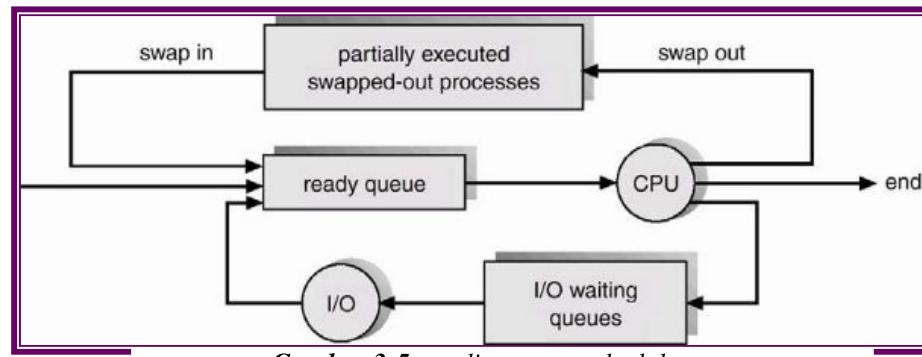
Gambar 5 : representasi penjadwalan proses

Penjadwal (Scheduler)

Terdapat dua bentuk penjadwal, yaitu:

- Longterm-Scheduler (job scheduler)*, menyeleksi proses-proses mana yang harus dibawa ke *ready queue*.
- Short-term Scheduler (CPU scheduler)*, memilih proses-proses yang siap untuk dieksekusi, dan mengakolakasikan CPU ke salah satu dari proses-proses tersebut.

Selain kedua jenis penjadwal diatas terdapat satu jenis penjadwal yang disebut dengan *medium-term scheduler*. Gambar 3-5 merupakan *medium-term scheduler*.



Gambar 3-5: medium-term scheduler

jarang (dalam detik atau menit), sehingga setiap proses dijadwal dengan lambat. *Long-term scheduler* digunakan untuk mengontrol tingkat multiprogramming.

Secara umum, proses dapat digambarkan sebagai :

- *I/O bound process*, yaitu proses-proses yang membutuhkan lebih banyak waktu untuk menjalankan I/O daripada melakukan komputasi, sehingga CPU burst yang dibutuhkan lebih singkat.
- *CPU bound process*, yaitu proses-proses yang membutuhkan lebih banyak waktu untuk melakukan komputasi daripada menjalankan I/O sehingga CPU burst yang dibutuhkan lebih lama.

Context Switch

Ketika CPU berpindah dari proses satu ke proses lainnya, sistem harus menyimpan status dari proses yang lama dan membuka state proses baru yang sudah disimpan. *Context switch* adalah proses penyimpanan status proses dan mengambil status proses yang baru pada saat terjadi switching. Pada saat terjadi perpindahan proses, sistem tidak bekerja. Waktu *context switch* tergantung pada perangkat keras yang digunakan.

A. PEMBUATAN PROSES

Terdapat dua operasi pada proses, yaitu pembuatan proses (*process creation*) dan penghentian proses (*process deletion*).

Pembuatan Proses

Ada beberapa aktivitas berkenaan dengan pembuatan proses, antara lain :

- a. Memberi identitas (nama) pada proses yang dibuat;
- b. Menyisipkan proses pada list proses atau tabel proses;
- c. Menentukan prioritas awal proses;
- d. Membuat PCB;
- e. Mengalokasikan resource awal bagi proses tersebut.

Ada beberapa kejadian yang menyebabkan pembuatan suatu proses baru, antara lain:

- a. Pada lingkungan batch sebagai tambahan atas pemberian job. Setelah

- b. Pada ingkungan interaktif, pada saat user baru saja logam;
- c. Sebagai tanggapan atas suatu aplikasi (seperti: mencetak file, sistem operasi dapat menciptakan proses yang akan mengelola pencetakan itu);
- d. Proses menciptakan proses lain (child).

Selama eksekusi, suatu proses mungkin akan membuat suatu proses yang baru. Proses tersebut dinamakan parent, sedangkan proses yang dibuat dinamakan child. Proses pembuatan proses anak membentuk pohon proses.

Pembagian sumber daya :

- Parent dan child membagi semua sumber daya yang ada
- Child menggunakan sebagian dari sumber daya yang digunakan parent
- Parent dan child tidak membagi sumber daya

Bentuk eksekusi :

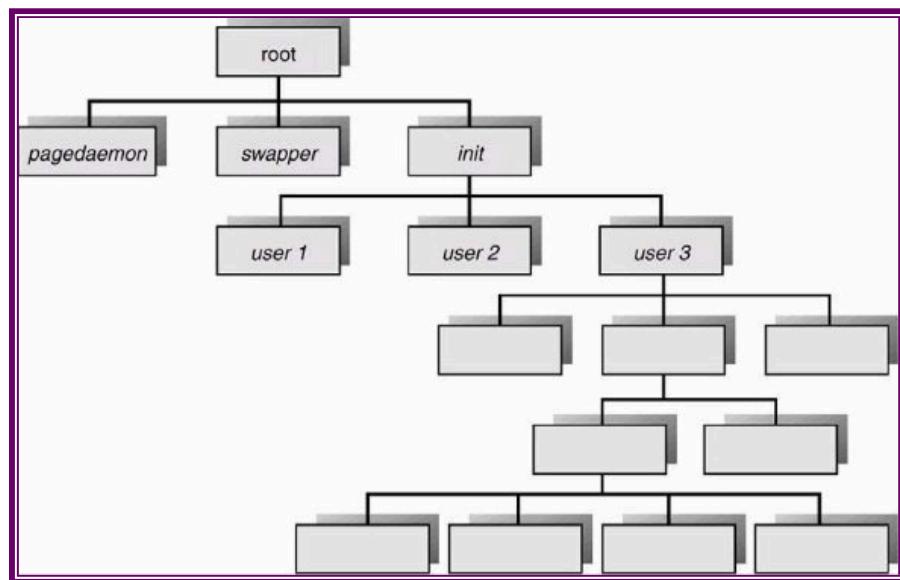
- Parent melanjutkan eksekusi beriringan dengan children.
- Parent menunggu hingga beberapa atau seluruh children selesai.

Bentuk ruang alamat :

- Child adalah duplikat dari proses parent.
- Child mempunyai program yang diambil dari dirinya.

Pada UNIX, parent akan membentuk child dengan menggunakan system call **fork**. Setelah pemanggilan **fork**, parent kembali berjalan secara pararel dengan child. Demikian pula, child dapat memanggil **fork** untuk membentuk child lainnya. Sistem call **exec** digunakan setelah system call **fork** mengganti alamat memori proses dengan program baru. Lain halnya dengan DOS, pada MS-DOS, system call akan memanggil binary file tertentu yang ada pada memori dan mengeksekusinya sebagai child. Parent akan running kembali setelah child selesai eksekusi. Dengan demikian parent dan child

pada Gambar 3-6.



Gambar 3-6: Bentuk pohon proses pada UNIX

Penghentian Proses

Suatu proses berhenti jika telah menyelesaikan pernyataan terakhir, dan meminta pada sistem operasi untuk menghapusnya dengan menggunakan system call **exit**. Proses mengembalikan semua data (output) ke parent proses melalui system call **wait**. Kemudian proses dihapus dari list atau tabel sistem, dilanjutkan dengan menghapus PCB.

Penghapusan proses ini akan menjadi sangat kompleks jika ternyata proses yang akan dihentikan tersebut membuat proses-proses yang lain. Pada beberapa sistem, proses-proses anak akan dihentikan secara otomatis jika proses induknya berhenti. Namun, ada beberapa sistem yang menganggap bahwa proses anak ini terpisah dengan induknya, sehingga proses anak tidak ikut dihentikan secara otomatis pada saat proses induk dihentikan.

system call **abort**. Proses anak dihentikan parent karena beberapa alasan, antara lain :

- Child mengalokasikan sumber daya melampaui batas
- Tugas child tidak dibutuhkan lebih lanjut
- Parent berhenti, karena system operasi tidak mengijinkan child untuk melanjutkan jika parent berhenti dan terminasi dilanjutkan.

Sumber

<https://garudacyber.co.id/artikel/650-konsep-dasar-dan-definisi-proses-pada-sistem-operasi>

http://eprints.dinus.ac.id/6356/1/3.2_Manajemen_Proses - Proses.pdf

www.google.com