

Mengidentifikasi Proses Sistem Operasi

ALMAUNAH 195120034

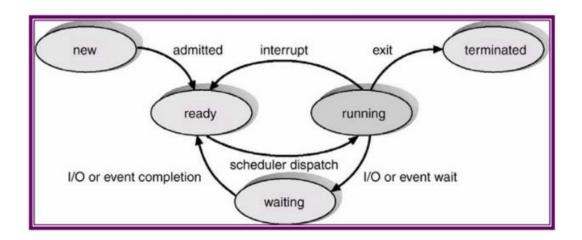
Fakultas Komputer Almaunah.student@umitra.ac.id

Isi Resume:

Pada multiprogramming, teknik penanganan proses adalah dengan menjalankan satu proses dan secara

cepat berpindah ke proses lainnya (bergiliran), sehingga menumbulkan 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.

Saat proses dieksekusi, akan terjadilah perubahan status. Status proses diartikan sebagai bagian dari

aktivitas proses yang sedang berlangsung pada saat itu. Status-status proses terdiri atas :

New: proses yang sedang dibuat

Running: proses yang sedang dijalankan

Waiting: proses sedang menunggu beberapa event (kejadian) yang akan terjadi

(seperti penyelesaian

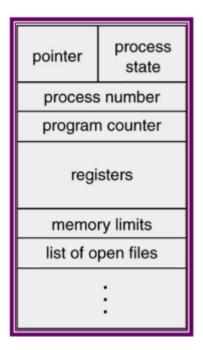
input output atau penerimaan sebuah signal / tanda)

Ready: proses sedang menunggu untuk diproses oleh prosesor

Terminated: proses telah selesai menjalankan tugasnya (selesai dieksekusi)

Setiap proses direpresentasikan / digambarkan oleh sistem operasi dengan menggunakan Process ControlBlock (PCB), seperti yang terlihat pada gambar di Samping Informasi yang terdapat pada setiap proses meliputi :

Status Proses: New, ready, running, waiting dan terminated.



Program Counter: Mengindikasikan alamat (address) dari perintah selanjutnya untuk dijalankan untuk proses ini.

CPU Registers: Register beragam dalam jumlah dan jenisnya tergantung pada rancangan arsitektur computer yang bersangkutan. Register tersebut terdiri dari accumulator, stack pointer, index register, general-purposes register, ditambah code information pada kondisi apa pun.

Selama Program Counter berjalan, status informasi harus disimpan pada saat terjadi interrupt.

Informasi Penjadwalan CPU : Informasi ini berisikan prisoritas dari sebuah proses, pointer ke antrian

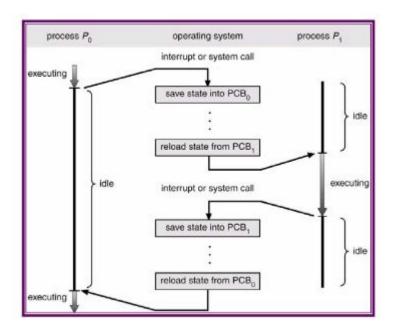
penjadwalan, dan beberapa parameter penjadwalan yang lainnya.

Informasi Manajemen Memori : Informasi ini berisikan nilai (basis) dan limit register, page table, atau

segment table tergantung dari sistem memori yang digunakan oleh sistem operasi.

Informasi Accounting: Informasi ini berisikan jumlah CPU dan real time yang digunakan, account

numbers, time limits, jumlah job maupun proses, dan lain sebagainya.



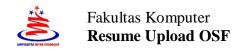
Informasi Status Input Output : Informasi ini berisikan deretan input output device (seperti tape driver)

yang dialokasikan untuk proses tersebut, deretan file yang dibuka, dan lain sebagainya.

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

Proses adalah sebuah konsep utama pada sistem operasi, sehingga masalah manajemen proses adalah masalah utama padaperancangan sistem operasi.

Proses adalah program yang sedang dieksekusi.



Proses adalah unit kerja terkecil yang secaraindividu 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 :

Independent, adalah program-program tersebut berdiri sendiri, tidak saling bergantung dan terpisah.

One program at any instant, adalah hanya satu proses yang dilayani oleh pemroses pada satu waktu.

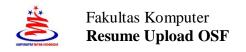
Pembuatan Proses

Suatu proses dapat membuat beberapa proses baru, melalui sistem pemanggilan pembuatan proses, selama jalur eksekusi. Pembuatan proses dinamakan induk proses, sebagaimana proses baru disebut anak dari proses tersbut. Tiap proses baru tersebut dapat membuat proses lainnya, membentuk suatu pohon proses (lihat Gambar 2-7).

Secara umum, suatu proses akan memerlukan sumber tertentu (waktu CPU, memori, berkas, perangkat I/O) untuk menyelesaikan tugasnya. Ketika suatu proses membuat sebuah subproses, sehingga subproses dapat mampu untuk memperoleh sumbernya secara langsung dari sistem operasi. Induk mungkin harus membatasi sumber diantara anaknya, atau induk dapat berbagi sebagian sumber (seperti memori berkas) diantara beberapa dari anaknya. Membatasi suatu anak proses menjadi subset sumber daya induknya mencegah proses apa pun dari pengisian sistem yang telalu banyak dengan menciptakan terlalu banyak subproses.

Sebagai tambahan pada berbagai sumber fisik dan logis bahwa suatu proses diperoleh ketika telah dibuat, data pemula (masukan) dapat turut lewat oleh induk proses sampai anak proses. Sebagai contoh, anggap suatu proses yang fungsinya untuk menunjukkan status sebuah berkas, katakan F1, pada layar terminal.

Ketika dibuat, akan menjadi sebagai sebuah masukan dari proses induknya, nama dari berkas F1, dan akan mengeksekusi menggunakan kumpulan data tersebut untuk memperoleh informasi yang diinginkan. Proses tersebut juga mendapat nama dari perangkat luar. Sebagian sistem operasi melewati sumber-sumber ke anak proses. Pada sistem tersebut, proses baru bisa mendapat dua berkas terbuka yang baru, F1



dan perangkat terminal dan hanya perlu untuk mentransfer data antara kedua berkas tersebut.

Ketika suatu proses membuat proses baru, dua kemungkinan ada dalam term eksekusi:

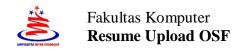
- 1. Induk terus menerus untuk mengeksekusi secara bersama-sama dengan anaknya.
- 2. Induk menunggu sampai sebagian dari anaknya telah diakhiri/terminasi.

Juga ada dua kemungkinan dalam term dari address space pada proses baru:

- 1. Anak proses adalah duplikat dari induk proses.
- 2. Anak proses memiliki program yang terisikan didalamnya.

Untuk mengilustrasikan implementasi yang berbeda ini, mari kita mempelajari sistem operasi UNIX. Dalam UNIX, tiap proses diidentifikasi oleh pengidentifikasi proses, yang merupakan integer yang unik. Proses baru dibuat oleh sistem pemanggilan fork system call. Proses baru tersebut terdiri dari sebuah copy ruang alamat dari proses aslinya (original). Mekanisme tersebut memungkinkan induk proses untuk berkomunikasi dengan mudah dengan anak proses. Kedua proses (induk dan anak) meneruskan eksekusi pada instruksi setelah fork dengan satu perbedaan: Kode kembali untuk fork adalah nol untuk proses baru (anak), sebagaimana proses pengidentifikasi non nol (non zero) dari anak dikembalikan kepada induk.

Umumnya, sistem pemanggilan execlp digunakan setelah sistem pemanggilan fork. Oleh satu dari dua proses untuk menggantikan proses ruang memori dengan program baru. Sistem pemanggilan execlp mengisi suatu berkas binary kedalam memori (menghancurkan gambar memori pada program yang berisikan sistem pemanggilan execlp) dan memulai eksekusinya. Dengan cara ini, kedua proses mampu untuk berkomunikasi, dan lalu untuk pergi ke arah yang berbeda. Induk lalu dapat membuat anak yang lebh banyak atau jika induk tidak punya hal lain untuk dilakukan ketika anak bekerja, induk dapat mengeluarkan sistem pemanggilan wait untuk tidak menggerakkan dirinya sendiri pada suatu antrian yang siap sampai anak berhenti. Program C ditunjukkan pada Gambar 2-10 mengilustrasikan sistem pemanggilan pada UNIX yang sebelumnya dijelaskan. Induk membuat anak proses menggunakan sistem pemanggilan fork(). Kini kita mempunyai dua proses yang berbeda yang menjalankan sebuah copy pada program yang sama. Nilai dari pid untuk anak proses adalah nol (zero): maka untuk induk adalah nilai integer yang lebih besar dari nol. Anak proses meletakkan ruang alamatnya dengan UNIX command/bin/ls (digunakan untuk mendapatkan pendaftaran directory) menggunakan sistem pemanggilan execlp(). Ketika anak proses selesai, induk proses menyimpulkan dari pemanggilan



untuk wait() dimana induk proses menyelesaikannya dengan menggunakan sistem pemanggilan exit().

Secara kontras, sistem operasi DEC VMS membuat sebuah proses baru dengan mengisi program tertentu kedalam proses tersebut, dan memulai pekerjaannya. Sistem operasi Microsoft Windows NT mendukung kedua model: Ruang alamat induk proses dapat di duplikasi, atau induk dapat menspesifikasi nama dari sebuah program untuk sistem operasi untuk diisikan kedalam ruang alamat pada proses baru.

Terminasi Proses

Sebuah proses berakhir ketika proses tersebut selesai mengeksekusi pernyataan akhirnya dan meminta sistem operasi untuk menghapusnya dengan menggunakan sistem pemanggilan exit. Pada titik itu, proses tersebut dapat mengembalikan data (keluaran) pada induk prosesnya (melalui sistem pemanggilan wait). Seluruh sumbersumber dari proses-termasuk memori fisik dan virtual, membuka berkas, dan penyimpanan I/O di tempatkan kembali oleh sistem operasi.

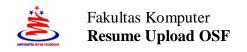
Ada situasi tambahan tertentu ketika terminasi terjadi. Sebuah proses dapat menyebabkan terminasi dari proses lain melalui sistem pemanggilan yang tepat (contoh abort). Biasanya, sistem seperti itu dapat dipanggil hanya oleh induk proses tersebut yang akan diterminasi. Bila tidak, pengguna dapat secara sewenang-wenang membunuh job antara satu sama lain. Catat bahwa induk perlu tahu identitas dari anaknya. Maka, ketika satu proses membuat proses baru, identitas dari proses yang baru diberikan kepada induknya.

Induk dapat menterminasi/ mengakhiri satu dari anaknya untuk beberapa alasan, seperti:

- Anak telah melampaui kegunaannya atas sebagaian sumber yang telah diperuntukkan untuknya.
- Pekerjaan yang ditugaskan kepada anak telah keluar, dan sistem operasi tidak memeperbolehkan sebuah anak untuk meneruskan jika induknya berakhir.

Untuk menentukan kasus pertama, induk harus memiliki mekanisme untuk memeriksa status anaknya. Banyak sistem, termasuk VMS, tidak memperbolehkan sebuah anak untuk ada jika induknya telah berakhir. Dalam sistem seperti ini, jika suatu proses berakhir (walau secara normal atau tidak normal), maka seluruh anaknya juga harus diterminasi. Fenomena ini, mengacu pada terminasi secara *cascading*, yang normalnya dimulai oleh sistem operasi.

Untuk mengilustrasikan proses eksekusi dan proses terminasi, kita menganggap bahwa, dalam UNIX, kami dapat mengakhiri suatu proses dengan sistem pemanggilan exit; proses induknya dapat menunggu untuk terminasi anak proses dengan menggunakan sistem pemanggilan wait. Sistem pemanggilan wait kembali ke pengidentifikasi proses dari anak yang telah diterminasi, maka induk dapat memberitahu kemungkinanan anak mana yang telah diterminasi. Jika induk



menterminasi. Maka, anaknya masih punya sebuah induk untuk mengumpulkan status mereka dan mengumpulkan statistik eksekusinya.

Hubungan Antara Proses

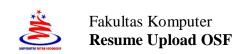
Sebelumnya kita telah ketahui seluk beluk dari suatu proses mulai dari pengertiannya, cara kerjanya, sampai operasi-operasinya seperti proses pembentukannya dan proses pemberhentiannya setelah selesai melakukan eksekusi. Kali ini kita akan mengulas bagaimana hubungan antar proses dapat berlangsung, misal bagaimana beberapa proses dapat saling berkomunikasi dan bekerja-sama.

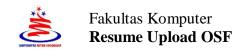
Proses yang Kooperatif

Proses yang bersifat simultan (*concurrent*) dijalankan pada sistem operasi dapat dibedakaan menjadi yaitu proses independent dan proses kooperatif. Suatu proses dikatakan independen apabila proses tersebut tidak dapat terpengaruh atau dipengaruhi oleh proses lain yang sedang dijalankan pada sistem.

Berarti, semua proses yang tidak membagi data apa pun (baik sementara/ tetap) dengan proses lain adalah independent. Sedangkan proses kooperatif adalah proses yang dapat dipengaruhi atau pun terpengaruhi oleh proses lain yang sedang dijalankan dalam sistem. Dengan kata lain, proses dikatakan kooperatif bila proses dapat membagi datanya dengan proses lain. Ada empat alasan untuk penyediaan sebuah lingkungan yang memperbolehkan terjadinya proses kooperatif:

- 1. Pembagian informasi: apabila beberapa pengguna dapat tertarik pada bagian informasi yang sama (sebagai contoh, sebuah berkas bersama), kita harus menyediakan sebuah lingkungan yang mengizinkan akses secara terus menerus ke tipe dari sumber-sumber tersebut.
- 2. Kecepatan penghitungan/ komputasi: jika kita menginginkan sebuah tugas khusus untuk menjalankan lebih cepat, kita harus membagi hal tersebut ke dalam subtask, setiap bagian dari subtask akan dijalankan secara parallel dengan yang lainnya. Peningkatan kecepatan dapat dilakukan hanya jika komputer tersebut memiliki elemen-elemen pemrosesan ganda (seperti CPU atau jalur I/O).
- 3. Modularitas: kita mungkin ingin untuk membangun sebuah sistem pada sebuah model modular-modular, membagi fungsi sistem menjadi beberapa proses atau threads.
- 4. Kenyamanan: bahkan seorang pengguna individu mungkin memiliki banyak tugas untuk dikerjakan secara bersamaan pada satu waktu. Sebagai contoh, seorang pengguna dapat mengedit, memcetak, dan meng-*compile* secara paralel.



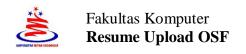


A. ID SECURITY

QWTD44112377-ASP-524414475

B. REFERENCE

- [1] O. M. Febriani and A. S. Putra, "Sistem Informasi Monitoring Inventori Barang Pada Balai Riset Standardisasi Industri Bandar Lampung," *J. Inform.*, vol. 13, no. 1, pp. 90–98, 2014.
- [2] A. S. Putra, "Paperplain: Execution Fundamental Create Application With Borland Delphi 7.0 University Of Mitra Indonesia," 2018.
- [3] A. S. Putra, "2018 Artikel Struktur Data, Audit Dan Jaringan Komputer," 2018.
- [4] A. S. Putra, "ALIAS MANAGER USED IN DATABASE DESKTOP STUDI CASE DB DEMOS."
- [5] A. S. Putra, "COMPREHENSIVE SET OF PROFESSIONAL FOR DISTRIBUTE COMPUTING."
- [6] A. S. Putra, "DATA ORIENTED RECOGNITION IN BORLAND DELPHI 7.0."
- [7] A. S. Putra, "EMBARCADERO DELPHI XE 2 IN GPU-POWERED FIREMONKEY APPLICATION."
- [8] A. S. Putra, "HAK ATAS KEKAYAAN INTELEKTUAL DALAM DUNIA TEKNOLOGY BERBASIS REVOLUSI INDUSTRI 4.0."
- [9] A. S. Putra, "IMPLEMENTASI PERATURAN PERUNDANGAN UU. NO 31 TAHUN 2000 TENTANG DESAIN INDUSTRI BERBASIS INFORMATION TECHNOLOGY."
- [10] A. S. Putra, "IMPLEMENTATION OF PARADOX DBASE."
- [11] A. S. Putra, "IMPLEMENTATION OF TRADE SECRET CASE STUDY SAMSUNG MOBILE PHONE."
- [12] A. S. Putra, "IMPLEMENTATION PATENT FOR APPLICATION WEB BASED CASE STUDI WWW. PUBLIKLAMPUNG. COM."
- [13] A. S. Putra, "IMPLEMENTATION SYSTEM FIRST TO INVENT IN DIGITALLY INDUSTRY."
- [14] A. S. Putra, "MANUAL REPORT & INTEGRATED DEVELOPMENT ENVIRONMENT BORLAND DELPHI 7.0."
- [15] A. S. Putra, "PATENT AS RELEVAN SUPPORT RESEARCH."
- [16] A. S. Putra, "PATENT FOR RESEARCH STUDY CASE OF APPLE. Inc."
- [17] A. S. Putra, "PATENT PROTECTION FOR APPLICATION INVENT."
- [18] A. S. Putra, "QUICK REPORT IN PROPERTY PROGRAMMING."
- [19] A. S. Putra, "REVIEW CIRCUIT LAYOUT COMPONENT REQUIREMENT ON ASUS NOTEBOOK."
- [20] A. S. Putra, "REVIEW TRADEMARK PATENT FOR INDUSTRIAL TECHNOLOGY BASED 4.0."
- [21] A. S. Putra, "TOOLBAR COMPONENT PALLETTE IN OBJECT ORIENTED PROGRAMMING."



- [22] A. S. Putra, "WORKING DIRECTORY SET FOR PARADOX 7."
- [23] A. S. Putra, "ZQUERY CONNECTION IMPLEMENTED PROGRAMMING STUDI CASE PT. BANK BCA Tbk."
 - [24] A. S. Putra, D. R. Aryanti, and I. Hartati, "Metode SAW (Simple Additive Weighting) sebagai Sistem Pendukung Keputusan Guru Berprestasi (Studi Kasus: SMK Global Surya)," in *Prosiding Seminar Nasional Darmajaya*, 2018, vol. 1, no. 1, pp. 85–97.
- [25] A. S. Putra and O. M. Febriani, "Knowledge Management Online Application in PDAM Lampung Province," in *Prosiding International conference on Information Technology and Business (ICITB)*, 2018, pp. 181–187.
- [26] A. S. Putra, O. M. Febriani, and B. Bachry, "Implementasi Genetic Fuzzy System Untuk Mengidentifikasi Hasil Curian Kendaraan Bermotor Di Polda Lampung," *SIMADA (Jurnal Sist. Inf. dan Manaj. Basis Data)*, vol. 1, no. 1, pp. 21–30, 2018.
- [27] A. S. Putra, H. Sukri, and K. Zuhri, "Sistem Monitoring Realtime Jaringan Irigasi Desa (JIDES) Dengan Konsep Jaringan Sensor Nirkabel," *IJEIS* (*Indonesian J. Electron. Instrum. Syst.*, vol. 8, no. 2, pp. 221–232.
- [28] D. P. Sari, O. M. Febriani, and A. S. Putra, "Perancangan Sistem Informasi SDM Berprestasi pada SD Global Surya," in *Prosiding Seminar Nasional Darmajaya*, 2018, vol. 1, no. 1, pp. 289–294.