MANAJEMEN PROSES

Pertemuan ke 4

Christy Atika Sari, M.Kom, M.CS

Outline

- Definisi Manajemen Proses
- Pengendalian Proses
- Diagram State 3 Keadaan
- Process Contol Block (PCB)
- Identifikasi Proses
- Diagram State 5 Keadaan
- Implementasi Proses
- Interupsi

Definisi Manajemen Proses [1]

- Definisi :
  - Adalah program yang sedang dieksekusi
- Unit terkecil yang secara individu memiliki sumber daya-sumber daya dan dijadwalkan oleh Sistem Operasi (SO)
- SO mengelola semua proses di sistem dan mengalokasikan sumber daya ke proses-proses sesuai dengan kebijaksanaan untuk memenuhi sasaran sistem
- Hal-hal berkaitan proses
  - Multiprogramming (Multitasking)
  - Multiprocessing
  - Distributed processing

Definisi Manajemen Proses [2]

- Definisi : Manajemen banyak proses pada satu pemroses
- Banyak proses yang dijalankan bersamaan, masing-masing proses mendapat bagian memori dan kendali sendiri
- Program yang dijalankan bersifat
  - Tidak bergantung (*Independent*)
    - Proses terpisah satu dari lainnya & tidak berpengaruh
  - Satu program pada satu saat (*one program at any instant*)
    - Pada satu waktu hanya satu proses yang dilayani pemroses, menggunakan *interleave* bukan *overlap* diantara program-program
- Oleh karena perpindahan dari satu proses ke proses dilakukan secara cepat bagi bagi pemakai seolah-olah bekerja secara paralel. Hal ini dikenal dengan paralel semu (*pseudoparallelism*)

Definisi Manajemen Proses [3]

- Definisi : Manajemen banyak proses di komputer *multiprocessor*
- Dengan kata lain komputer dengan banyak pemroses di satu sistem komputer dengan masing-masing pemroses melakukan pemrosesan secara independen
- Contoh SO yang mendukung : Windows NT, UNIX, LINUX

Definisi Manajemen Proses [4]

- MAnajemen banyak proses yang dieksekusi di banyak sistem komputer yang tersebar (terdistribusi).
- Contoh : MACH, AMOEBA

Pengendalian Proses [1]

- Kebutuhan utama pengendalian proses oleh SO dapat dinyatakan dengan mengacu ke proses yaitu
  - Saling melanjutkan (*interleave*)
  - Mengikuti kebijaksanaan tertentu
  - Mendukung komunikasi antar proses dan penciptaan proses

Pengendalian Proses [2]

- Dikatakan *interleave* (bersambung/ melanjutkan) maksudnya pemroses mengeksekusi satu proses setiap saat dan secara cepat beralih ke proses lainnya secara bergiliran.
- SO harus *interleave* (saling melanjutkan) eksekusi proses-proses agar memaksimumkan penggunaan pemroses sambil masih memberi waktu tanggap yang memadai

Pengendalian Proses [3]

- SO harus mengalokasikan sumber daya ke proses-proses mengikuti kebijaksanaan yang ditentukan (misal suatu aplikasi memiliki prioritas lebih tinggi) sambil menghindari *deadlock*

Pengendalian Proses [4]

- SO harus mendukung komunikasi antar proses dan penciptaan proses oleh pemakai sehingga membantu menstrukturkan aplikasi.
  - Jadi pada sistem dengan banyak **proses aktif**, proses-proses pada satu saat berada dalam beragam tahap eksekusinya.
    - Proses mengalami **beragam *state*** selama siklus hidupnya sebelum berakhir dan keluar dari sistem.
  - SO harus mengetahui *state* masing-masing proses dan merekam semua perubahan yang terjadi secara dinamis.
    - Informasi ini untuk penjadwalan dan memutuskan alokasi sumber daya

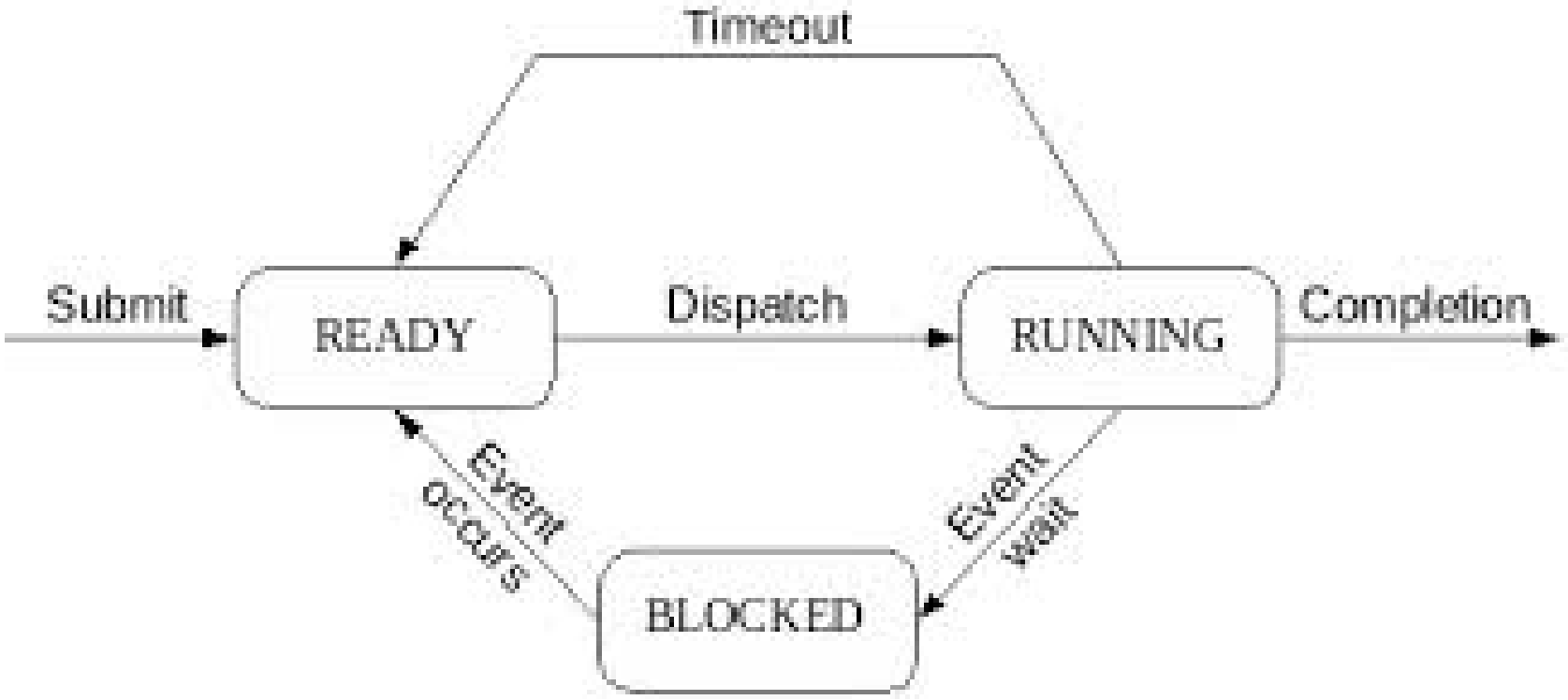
DIAGRAM STATE PROSES

1. State dasar (3 keadaan)
2. State lanjut (5 keadaan)

State Diagram 3 Proses [1]

- Running : pemroses sedang mengeksekusi innstruksi proses tersebut
- Ready : proses siap dieksekusi, tapi pemroses tidak tersedia untuk eksekusi proses ini
- Blocked : proses menunggu kejadian (event) untuk melengkapi tugasnya

State Diagram 3 Proses [2]



State Diagram 3 Proses [3]

- Proses baru diciptakan berada pada state *ready*
- Proses dari *running* menjadi *blocked* karena sumberdaya yang diminta belum tersedia atau meminta layanan perangkat masukan/ keluaran (I/O) sehingga menunggu kejadian yang muncul. Proses ini dikenal dengan *event wait.*
- Proses dari *running* jadi *ready* karena penjadwal memutuskan eksekusi proses lain oleh karena jatah waktu telah habis (*timeout*).
- Proses dari *blocked* jadi *ready* karena sumber daya yang diminta tersedia atau layanan I/O selesai/ terpenuhi. Proses ini dikenal *event occur*
- Proses dari *ready* jadi *running* karena penjadwal memutuskan untuk mengeksekusi proses tersebut.

Process Control Block (PCB) [1]

- Tiap proses digambarkan dalam sistem operasi oleh sebuah *process control block* (PCB) - juga disebut sebuah *control block*.
- PCB berisikan banyak bagian dari informasi yang berhubungan dengan sebuah proses yang spesifik, yaitu :
  1. Keadaan proses: Keadaan mungkin, *new*, *ready*, *running*, *waiting*, *halted*, dan juga banyak lagi.
  2. *Program counter*: *Counter* mengindikasikan address dari perintah selanjutnya untuk dijalankan untuk proses ini.
  3. CPU register: Register bervariasi dalam jumlah dan jenis, tergantung pada rancangan komputer. Register tersebut termasuk accumulator, index register, stack pointer, general-puposes register, ditambah code information pada kondisi apa pun. Besertaan dengan program counter, keadaan/ status informasi harus disimpan ketika gangguan terjadi, untuk memungkinkan proses tersebut berjalan/ bekerja dengan benar setelahnya.

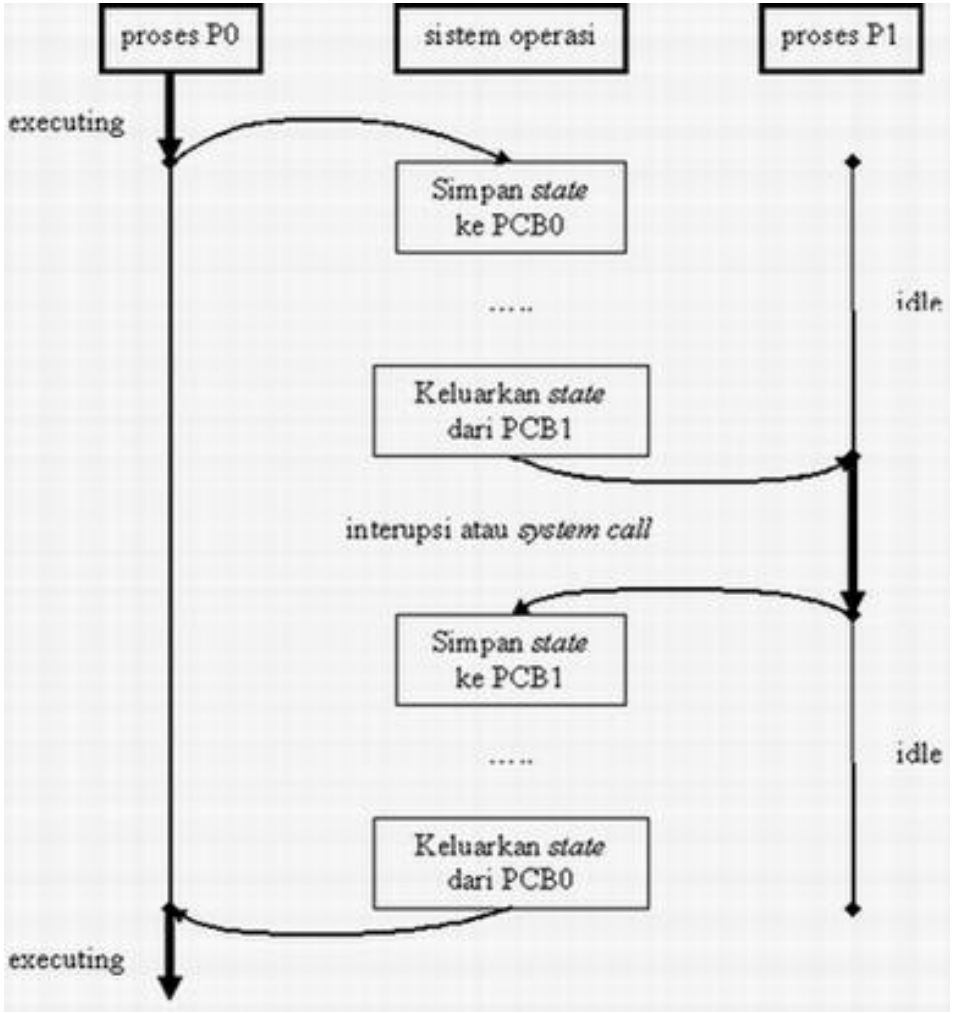
Process Control Block (PCB) [2]

4. Informasi managemen memori: Informasi ini dapat termasuk suatu informasi sebagai nilai dari dasar dan batas register, tabel page/ halaman, atau tabel segmen tergantung pada sistem memori yang digunakan oleh sistem operasi.
5. Informasi pencatatan: Informasi ini termasuk jumlah dari CPU dan waktu riil yang digunakan, batas waktu, jumlah akun, jumlah job atau proses, dan banyak lagi.
6. Informasi status I/O: Informasi termasuk daftar dari perangkat I/O yang di gunakan pada proses ini, suatu daftar open berkas dan banyak lagi.

- PCB hanya berfungsi sebagai tempat menyimpan/ gudang untuk informasi apa pun yang dapat bervariasi dari prose ke proses.

Process Control Block (PCB) [3]

Ilustrasi PCB



Ilustrasi di Register CPU

Informasi pada PCB [1]

- SO memerlukan banyak informasi mengenai proses guna pengelolaan proses
- Informasi ini ada di PCB
- Struktur datanya menyimpan informasi lengkap mengenai proses sehingga dapat terjadi siklus hidup proses
- Informasi di PCB dikelompokkan
  - Informasi identifikasi proses
  - Informasi status proses
  - Informasi kendali proses

Informasi pada PCB [2]

- Berkaitan dengan identitas proses yang unik
- Dengan identifier ini proses dikaitkan ke tabel-tabel lain
- Identifiernya adalah numerik yang meliputi
  - Identifier proses
  - Identifier proses yang menciptakan
  - Identifier pemakai

Informasi pada PCB [3]

- Informasi ini esensinya terdiri dari register-register pemroses.
- Saat proses berstatus *running*, informasi-informasi ini berada di register-register.
- Ketika proses diinterupsi semua informasi register harus disimpan agar dapat dikembalikan saat proses dieksekusi kembali
  - Jumlah dan ragam register bergantung pada arsitektur komputernya

Informasi pada PCB [4]

- Informasi ini esensinya terdiri dari register-register pemroses.
- Saat proses berstatus *running* informasi-informasi ini berada di register-register.
- Saat proses diinterupsi semua informasi register harus disimpan agar dapat dikembalikan saat proses dieksekusi kembali
- Jumlah dan ragam register yang terlibat bergantung pada arsitektur komputer

Informasi pada PCB [5]

- Adalah informasi-informasi lain yang diperlukan SO untuk mengendalikan dan koordinasi beragam proses aktif

Identifikasi Proses

Identifikasi Proses
Identifier
Identifier numerik yang meliputi ■ Identifier proses ■ Identifier proses yang menciptakan ■ Identifier pemakai

Informasi Status Pemroses

Informasi Status Pemroses
Register-register yang terlihat pemakai
Register-register yang dapat ditunjuk instruksi bahasa assembly untuk diproses pemroses
Register-register kendali dan status
Register-register yang digunakan untuk mengendalikan operasi pemroses, a.l.: ■ Program counter ■ PSW, dsb.
Pointer stack
Tiap proses mempunyai satu stack atau lebih. Stack digunakan untuk parameter atau alamat prosedur pemanggil dan system call. Pointer stack menunjuk posisi paling atas dari stack

Informasi Kendali Pemroses [1]

Informasi Kendali Pemroses
Informasi penjadwalan dan status
Informasi-informasi yang dipakai untuk menjalankan fungsi penjadwalan a.l : ■ **Status proses**. Mendefinisikan status proses (*running,ready,block, dsb*) ■ **Prioritas**. Menjelaskan prioritas proses ■ **Informasi berkaitan penjadwalan**. Informasi ini seperti lama menunggu, lama proses terakhir dieksekusi dsb. ■ **Kejadian (*Event*).** Identitas kejadian yang ditunggu proses
Penstrukturan data
Suatu proses dapat dikaitkan dengan proses lain dalam satu antrian atau *ring*, atau struktur lainnya. PCB harus memiliki *pointer* untuk mendukung struktur ini.
Komunikasi antar proses
Beragam flag, sinyal dan pesan dapat diasosiasikan dengan komunikasi antara dua proses yang terpisah. Informasi ini disimpan dalam PCB

Informasi Kendali Pemroses [2]

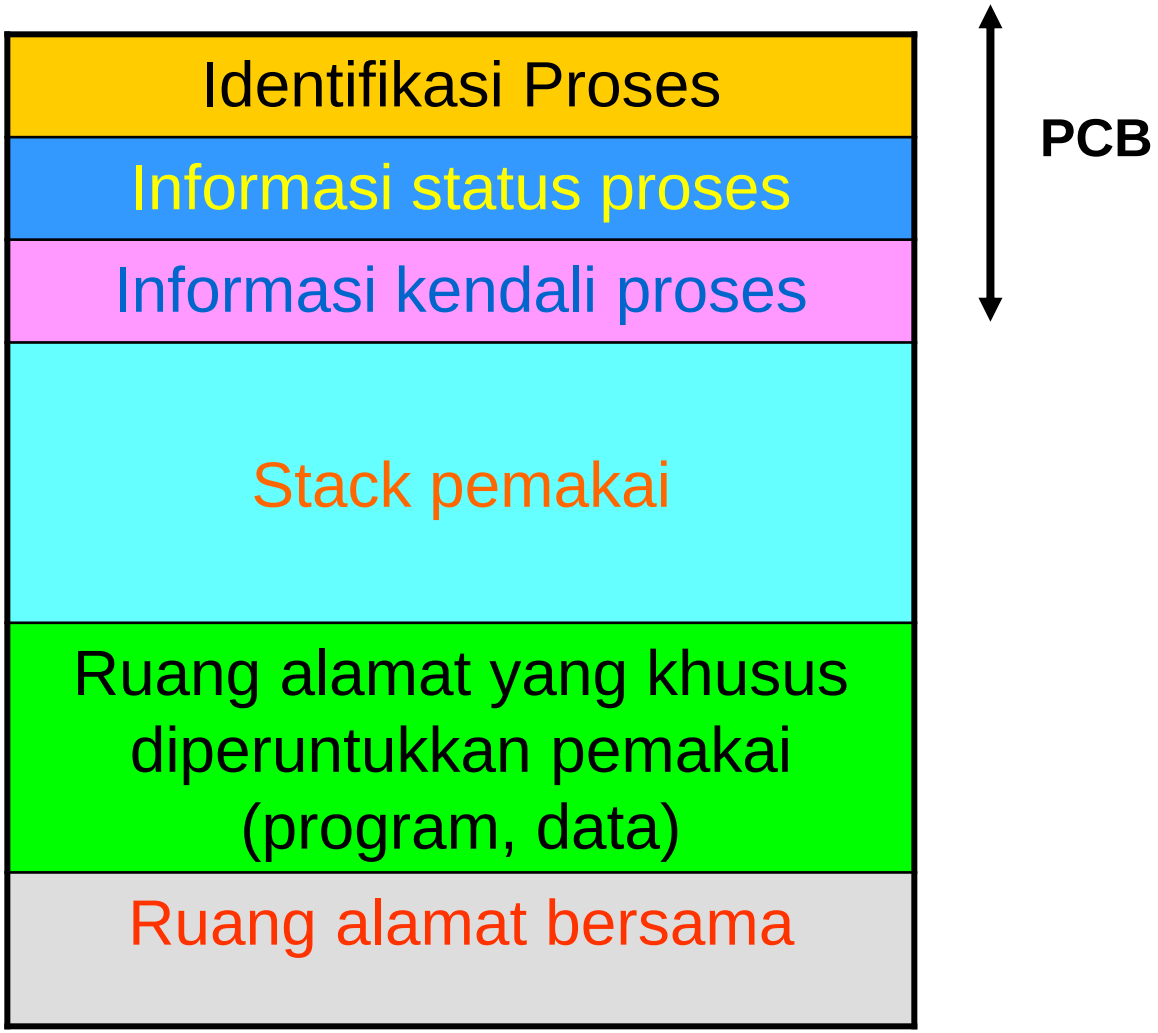
ELEMEN-ELEMEN PCB (lanjut)

Informasi Kendali Pemroses (lanjut)
Kewenangan proses
Proses dapat mempunyai kewenangan berkaitan dengan memori dan tipe instruksi yang dapat dijalankan
Manajemen memori
Bagian ini berisi pointer ke tabel segmen atau *page* yang menyatakan memori virtual proses
Kepemilikan dan utilisasi sumber daya
Sumber daya yang dikendalikan proses harus diberi tanda, misalnya : ■ Berkas yang dibuka ■ Pemakaian pemroses ■ Pemakaian sumberdaya lainnya Informasi ini diperlukan oleh penjadwal

Identifikasi Proses [1]

- Citra proses mempunyai struktur :
  - PCB
  - *Stack* pemakai (*User Stack*)
  - Ruang alamat proses eklusif
  - Ruang alamat yang dipakai bersama proses lain
- Implementasi penempatan citra proses yang sesungguhnya bergantung pada skema manajemen memori yang digunakan dan organisasi struktur kendali sistem operasi

Identifikasi Proses [2]



Identifikasi Proses [3]

- SO dalam mengelola proses dapat melakukan operasi-operasi terhadap proses.
- Operasi-operasi terhadap proses a.l. :
  - Penciptaan proses (*create process*)
  - Penghancuran/terminasi proses (*destroy a process)*
  - Penundaan proses (*suspend a process*)
  - Pelanjutan kembali proses (*resume process*)
  - Pengubahan prioritas proses
  - Memblok proses
  - Membangunkan proses
  - Menjadwalkan proses
  - Memungkinkan proses berkomunikasi dengan proses lain

Identifikasi Proses [4]

- Penciptaan proses melibatkan banyak aktivitas, yaitu
  - Menamai (memberi identitas) proses
  - Menyisipkan proses pada senarai proses atau tabel proses
  - Menentukan prioritas awal proses
  - Menciptakan PCB
  - Mengalokasikan sumberdaya awal bagi proses
- Ketika proses baru ditambahkan, SO membangun struktur data untuk mengelola dan alokasi ruang alamat proses itu. Aksi ini berkaitan dengan proses baru.

Identifikasi Proses [5]

- Kejadian yang dapat menyebabkan penciptaan proses a.l. :
  - Pada lingkungan *batch*, sebagai tanggapan atas pemberian suatu kerja (*job*)
  - Pada lingkungan interaktif, ketika pemakai baru berusaha logon
  - Sebagai tanggapan suatu aplikasi, seperti permintaan pencetakan file, SO dapat menciptakan proses yang akan mengelola pencetakan itu
  - Proses menciptakan proses lain (proses anak)
    - Proses yang menciptakan *child process* disebut proses induk (*parent process*)
    - *Child process* dapat menciptakan proses baru.
    - Proses-proses dapat membentuk pohon hirarki proses

Identifikasi Proses [6]

Penyebab Penciptaan	Deskripsi
Terdapat *batch job* baru	SO dengan kendali *batch job*, setelah menciptakan proses baru, kemudian melanjutkan membaca job selanjutnya
Satu pemakai interaktif *logon*	Seorang pemakai pada satu terminal sedang melakukan logon ke sistem
SO menciptakan proses untuk memberi layanan	SO menciptakan proses untuk memenuhi satu fungsi pada program pemakai, tanpa mengharuskan pemakai menunggu
Proses menciptakan proses anak	Untuk mencapai modularitas atau mengeksploitasi kongkuransi, program pemakai memerintahkan pembuatan sejumlah proses

Identifikasi Proses [3]

- **UNIX**
  - Proses baru diciptakan dengan *system call fork* (SCF). SCF menciptakan kopian proses pemanggil (induk) yang identik. Setelah panggilan *fork*, proses induk melanjutkan berjalan bersama proses anak secara paralel. Proses induk dapat kembali melakukan *fork* untuk menciptakan proses-proses anak yang baru. Proses anakpun dapat mengeksekusi *fork*, sehingga dapat terbentuk pohon hirarki proses.
- **MSDOS**
  - *System call* yang ada di MSDOS adalah me*load* file binerke memori dan mengeksekusi sebagai proses anak. Berbeda dengan UNIX, MSDOS panggilan ini menunda (menonaktifkan) proses induk sampai proses anak menyelesaikan eksekusi. Proses induk dan proses anak tidak berjalan secara paralel

Identifikasi Proses [3]

- Melibatkan pembebasan proses dari sistem, yaitu
  - Sumber daya-sumber daya yang dipakai dikembalikan
  - Proses dihancurkan dari senarai atau tabel sistem
  - PCB dihapus (ruang memori PCB dikembalikan ke *pool* bebas)
- Penghancuran lebih rumit bila proses telah menciptakan proses-proses lain. Terdapat dua pendekatan, yaitu
  - Pada beberapa sistem, proses-proses turunan dihancurkan saat proses-proses induk dihancurkan secara otomatis
  - Beberapa sistem lain menganggap proses anak independen terhadap proses induk sehingga proses anak tidak secara otomatis dihancurkan saat proses induk dihancurkan

Identifikasi Proses [3]

Penyebab Terminasi	Deskripsi
Selesainya proses secara normal	Proses mengeksekusi panggilan layanan SO untuk menandakan bahwa proses telah berjalan secara lengkap
Batas waktu telah terlewati	Proses telah berjalan melebihi batas waktu total yang dispesifikasikan. Terdapat banyak kemungkinan untuk tipe waktu yang diukur, termasuk waktu total yang dijalani (*walk clock time*), jumlah waktu yang dipakai untuk eksekusi, dan jumlah waktu sejak pemakai terakhir kali memberi masukan (pada proses interaktif)
Memori tidak tersedia	Proses memerlukan memori lebih banyak daripada yang dapat disediakan sistem
Pelanggaran terhdap batas memori	Proses mencoba mengakses lokasi memori yang tidak diijinkan diakses

Identifikasi Proses [3]

Penyebab Terminasi	Deskripsi
Terjadi kesalahan karena pelanggaran proteksi	Proses berusaha menggunakan sumberdaya atau file yang tidak diijinkan dipakainya, atau proses mencoba menggunakannya tidak untuk peruntukkannya, seperti menulis file *read-only*
Terjadi kesalahan aritmatika	Proses mencoba perhitungan terlarang, seperti pembagian dengan nol, atau mencoba menyimpan angka yang lebih besar daripada yang dapat diakomodasi oleh perangkat keras
Waktu telah kadaluarsa	Proses telah menunggu lebih lama daripada daripada maksimum yang ditentukan untuk terjadinya suatu kejadian spesifik
Terjadi kegagalan I/O	Kesalahan muncul pada input atu output, seperti ketidakmampuan menemukan file, kegagalan read atau write setelah sejumlah maksimum percobaan yang ditentukan (misal: area rusak pada tape, atau operasi tidak valid spt membaca dari line printer)

Identifikasi Proses [3]

Penyebab Terminasi	Deskripsi
Instruksi yang tidak benar	Proses berusaha mengeksekusi instruksi yang tidak ada (sering sebagai akibat percabangan ke daerah data dan berusaha mengeksekusi data itu)
Terjadi usaha memakai instruksi yang tidak diizinkan	Proses berusaha menggunakan instruksi yang disimpan untuk SO
Kesalahan penggunaan data	Bagian data adalah tipe yang salah atau tidak diinisialisasi
Diintervensi oleh SO atau operator	Untuk suatu alasan, operator atau SO mengakhiri proses (misal : terjadi *deadlock*)
Berakhirnya proses induk	Ketika induk berakhir, SO mungkin dirancang secara otomatis mengakhiri semua anak proses dari induk tsb.
Atas permintaan dari proses induk	Proses induk biasanya mempunyai otoritas mengakhiri suatu anak proses

Identifikasi Proses [3]

- Penundaan *(suspension)* adalah operasi penting dan telah diterapkan dengan beragam cara. Penundaan biasanya berlangsung singkat. Penundaan sering dilakukan sistem untuk memindahkan proses-proses tertentu guna mereduksi beban sistem selama beban puncak.

- Proses yang ditunda *(suspended process)* tidak berlanjut sampai proses lain me-resume. Untuk jangka panjang, sumber daya-sumber daya proses dibebaskan (dilucuti). Keputusan membebaskan sumber daya-sumber daya bergantung sifat masing-masing sumber daya. Memori utama seharusnya segera dibebaskan begitu proses tertunda agar dapat dimanfaatkan proses lain. *Resuming* (pengaktifan kembali) proses yaitu menjalankan proses dari titik (instruksi) dimana proses ditunda.

Identifikasi Proses [3]

- Operasi *suspend d*an *resume* penting, sebab:
  - Jika sistem berfungsi secara buruk dan mungkin gagal maka proses-proses dapat di *suspend* agar *di-resume s*etelah masalah diselesaikan.

    Contoh :

    Pada proses pencetakan, bila tiba-tiba kertas habis maka proses *di-suspend. S*etelah kertas dimasukkan kembali, proses pun dapat *di-resume.*
  - Pemakai vang ragu/kawatir mengenai hasil proses dapat men-suspend proses [bukan membuang *(abort)* proses]. Saat pemakai yakin proses akan berfungsi secara benar maka dapat *me-resume (melanjutkan* kembali di instruksi saat di-suspend) proses yang di-suspend
  - Sebagai tanggapan terhadap fluktuasi jangka pendek beban sistem, beberapa proses dapat di-suspend dan di- *resume s*aat beban kembali ke tingkat normal.

State Diagram 5 Keadaan [1]

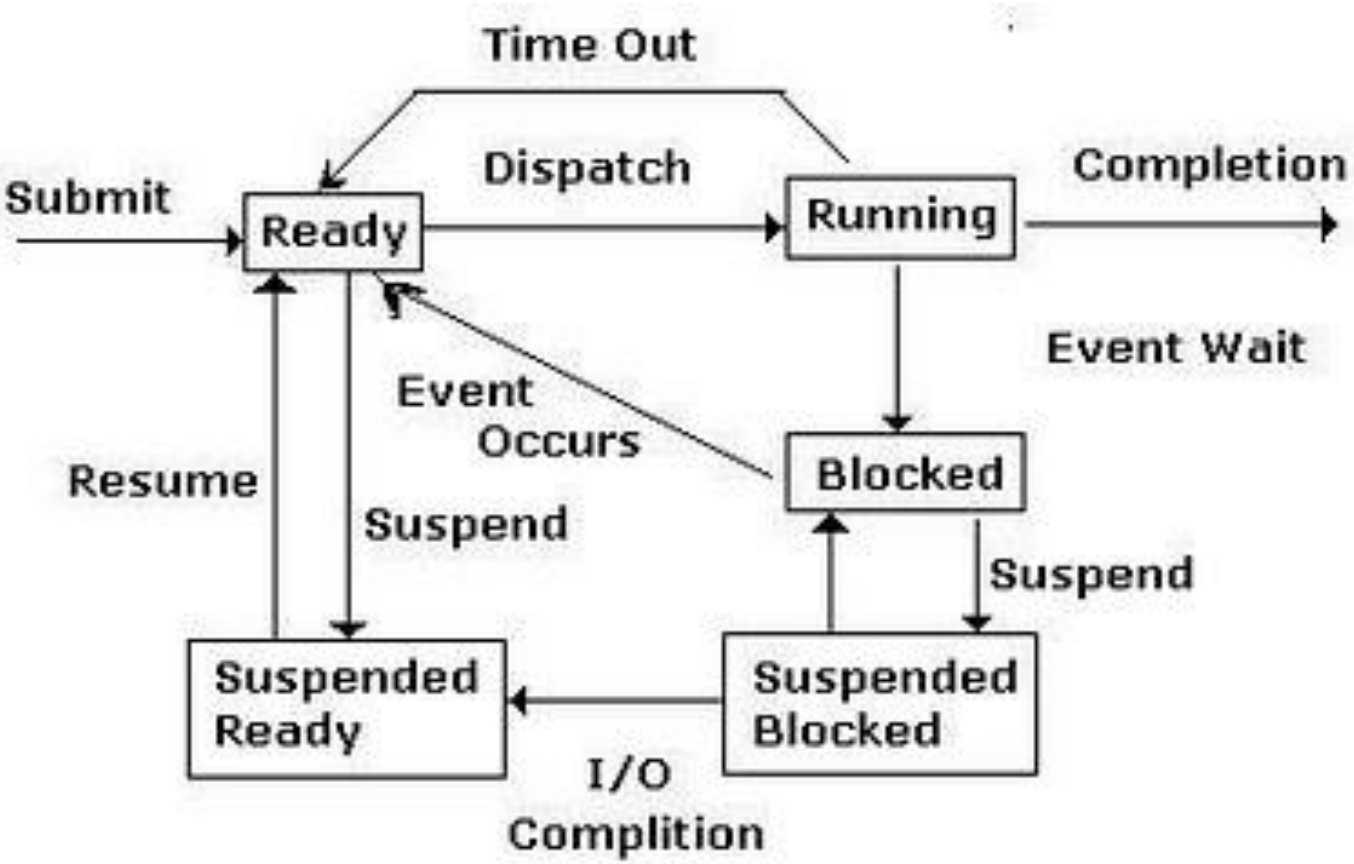


Diagram 5 State Proses

Gambar menunjukkan modifikasi diagram *state* dengan memasukkan kejadian *suspend* dan *resume*

State Diagram 5 Keadaan [2]

- Dua *state* baru dimasukkan sehingga membentuk diagram 5 *state,* yaitu
  1. Suspendedready.
  2. Suspendedblocked.
- Penundaan dapat diinisialisasi oleh proses itu sendiri atau proses lain.
  - Pada sistem *monoprocessor,* proses *running* dapat men-suspenddirinya sendiri karena tak ada proses lain yang juga *running y*ang dapat memerintahkan suspend.
  - Pada sistem *multiprocessor,* proses *running* dapat *di-suspend proses running* lain pada pernroses berbeda. Proses *ready* hanya dapat *di-suspend o*leh proses lain.
- Pada proses *blocked* terdapat transisi menjadi *suspendedblocked.* Pilihan ini dirasa aneh Apakah tidak cukup menunggu selesainya operasi masukan/keluaran atau kejadian yang membuat proses *ready* atau *suspendedready? B*ukankah *state blocked, readyblocked suspendedblocked* sama-sama tidak mendapatjatah waktu pernroses? Kenapa dibedakan?

State Diagram 5 Keadaan [3]

- Karena penyelesaian operasi masukanlkeluaran bagi proses *blocked* mungkin tak pernah terjadi atau dalam waktu tak terdefinisikan sehingga lebih baik *di-suspend agar* sumber dayasumber daya yang dialokasikan untuk proses tersebut dapat digunakan proses-proses lain. Untuk kondisi ini, lebih baik sumber daya-sumber daya yang dipegang proses yang berkondisi seperti ini dipakai proses-proses lain.
- Proses *blocked di-suspend s*istem atau secara manual menjadi *suspendedblocked.* Bila akhirnya operasi masukan/keluaran berakhir maka segera proses *suspendedblocked* mengalami transisi. Karena *resume* dan *suspend* mempunyai prioritas tinggi maka transisi sewra dilakukan. Susnenddan *resume d*anat digunakan untuk menveimbanekan beban sistem saat mengalami lonjakan di atas normal

Implementasi Proses

Tabel-tabel Untuk Proses

- Tiap proses mempunyai state yang perlu diperhatikan sistem operasi. Sistem operasi mencatat state proses dengan beragam tabel atau senarai, antara lain:
  1. Tabel infortnasi manajemen mernori.
  2. Tabel informasi manajemen masukan/keluaran.
  3. Tabel inforinasi sistern file.
  4. Tabel proses.
- Keempat tabel saling berhubungan

Tabel Informasi Manajemen Memori

- Tabel. informasi manajemen memori untuk meniaga keutuhan memori utama dan memori sekunder. Tabel ini memuat informasi berikut:
  - Alokasi memori utama yang dipakai proses.
  - Alokasi memori sekunder yang dipakai proses (bila menggunakan manaJemen memori dengan *swapping).*
- Atribut segmen memori utama dan sekunder.
- Informasi-informasi lain yang digunakan untuk pengelolaan memori,

Tabel Informasi Manajemen I/O

- Tabel ini untuk mengelola perangkat masukanlkeluaran. Pada satu saat, perangkat masukan/keluaran digunakan proses tertentu, perlu dijaga agar proses lain tidak memakainya. Sistern operasi perlu mengetahui status operasi masukan/keluaran dan lokasi memori utama yang digunakan untuk *transfer* data.

Tabel Informasi Sistem File

- Tabel ini berisi informasi mengenai ekstensi file, lokasi pada memori sekunder, status saat itu dan menyimpan atribut-atribut file lainnya.

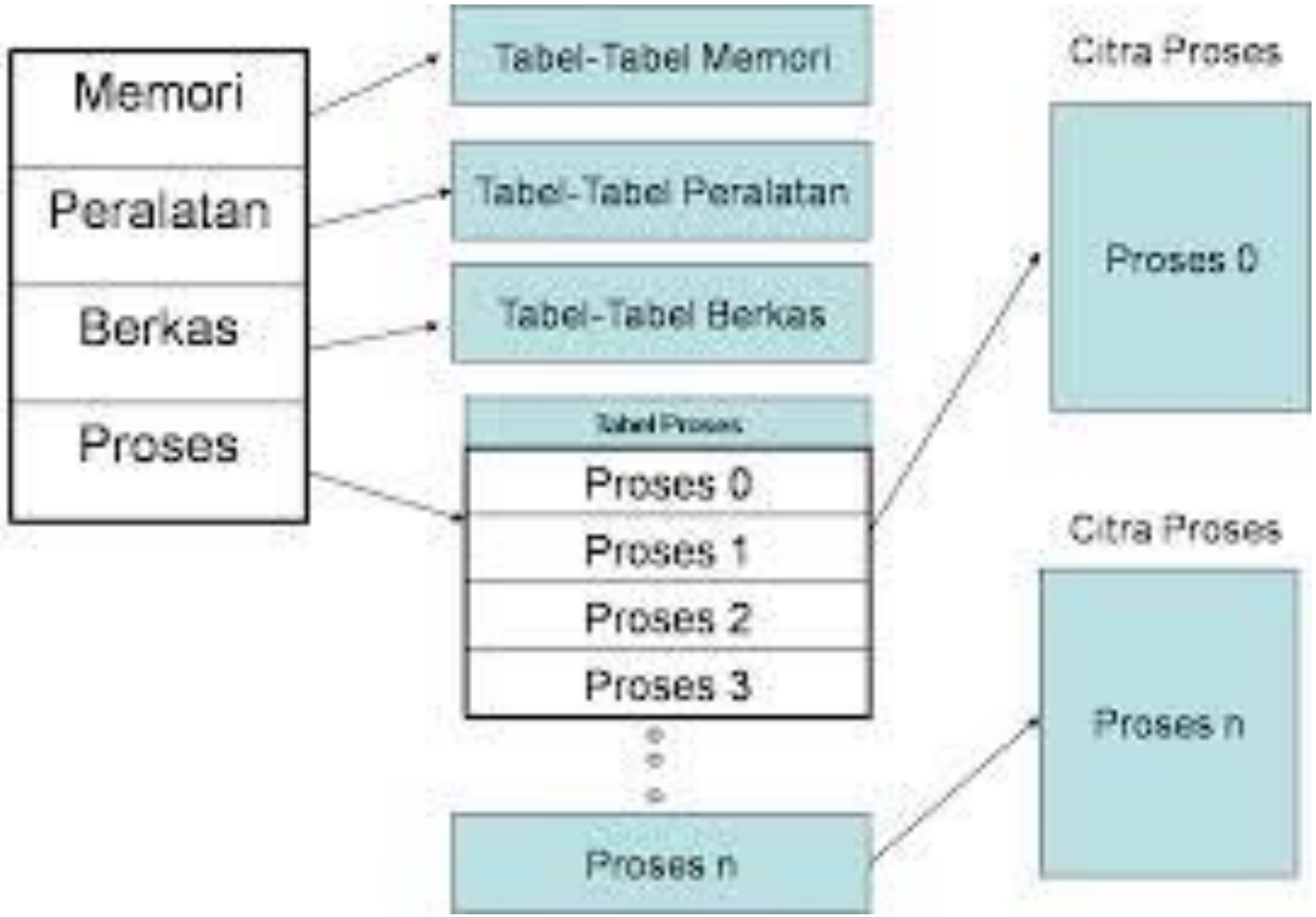
Tabel Proses

- Tabel proses mengelola informasi proses di sistem operasi, lokasinya di memori. Tabel juga berisi status dan atribut-atribut proses yang lain.
- Proses ditempatkan di memori utama di lokasi tertentu, proses mempunyai satu ruang alamat tersendiri. Ruang alamat yang digunakan proses disebut citra proses *(process image)* karena selain seluruh kode biner program, proses ditambahi atribut-atribut lain berkaitan penernpatannya pada suatu lokasi mernori dan status eksekusi pada saat itu.

Tabel Elemen Citra Proses

No	Elemen Citra	Keterangan
1	Data Pemakai	Bagian yg dpt memodifikasi berupa data program, daerah stack pemakai
2	Program Pemakai	Program biner yg dieksekusi
3	Stack Sistem	Digunakan untuk menyimpan parameter dan alamat pemanggilan untuk proedur dan system calls
4	PCB (Program Control Block)	Berisi informasi yg diperlukan oleh sistem operasi dalam pengendalian proses

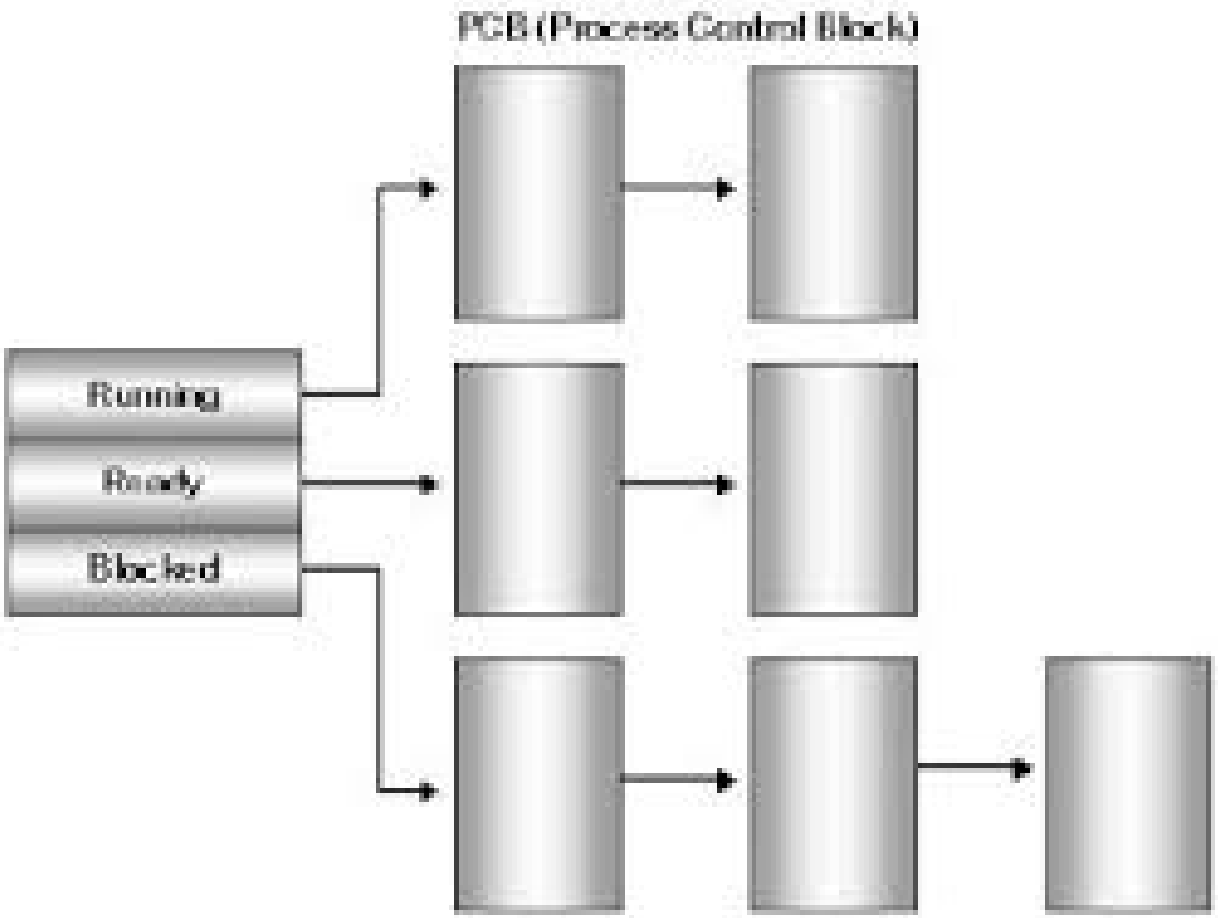
Elemen Citra Proses



PCB dan Senarai Proses

- PCB berperan penting di sistem operasi. Tiap PCB berisi informasi mengenai proses yang diperlukan sistern operasi. PCB dibaca dan/atau dimodifikasi rutin sistern operasi seperti penjadwalan, alokasi sumber daya, pemrosesan interupsi, *monitoring* dan analisis kinerja. Kumpulan PCB mendefinisikan state sistern operasi.
- Untuk menyatakan senarai proses di sistern operasi dibuat senarai KB.

Senarai PCB [1]



Senarai PCB [2]

- Gambar di atas memperlihatkan hanya satu PCB berada di senarai *running.* PCB ini menyatakan proses yang saat itu sedang dicksekusi pernroses sehingga hanya satu proses yang *running.* Tentu saja ini tidak berlaku untuk *multiprocessing* yang dapat mengeksekusi lebih dari satu proses sekaligus.
- Proses-proses *ready* digambarkan dengan PCB proses-proses di senarai *ready.* Proses-proses menunggu dijadwalkan untuk dieksekusi pemroses. Proses yang dijadwalkan dieksekusi (yaitu mengalami transisi dari *state ready* menjadi *running)* maka PCB-nya dipindah dari senarai *ready* ke senarai *running.*

Senarai PCB [3]

- Proses *running* (PCB-nya berada di senarai *running)* dipindah sesuai *state* yang dialami proses itu, sebagai berikut:
  - Bila proses berakhir (selesai) maka dijalankan operasi terminasi sehingga PCB-nya tak ada lagi.
  - Bila proses di-blocked karena menunggu alokasi sumber daya maka PCB-nya dipindah ke senarai *blocked.*
  - Bila proses dijadwalkan habisjatah waktu eksekusinya rnaka PCB-nya dipindahkan ke senarai *ready.*
- Proses yang sedang *blocked* berpindah menjadi *ready* bila sumber daya yang ditunggu telah teralokasi untuknya. Untuk itu PCB-nya dipindahkan ke senarai *ready.*

Pengaksesan Informasi pada PCB [1]

- Rutin-rutin sistern operasi perlu mengakses informasi di PCB. Tiap proses dilengkapi ID unik yang digunakan sebagai indeks (penunjuk) ke tabel untuk mengambil PCB.
- Kesulitan bukan pada mekanisme pengaksesan, tapi masalah proteksi terhadap PCB. Dua masalah utama proteksi terhadap PCB, yaitu:
  1. *Bug* (kesalahan pernrograrnan) pada rutin tunggal misalnya *interrupt handler* dapat merusak PCB sehingga dapat berakibat menghancurkan kernarnpuan sistem mengelola proses-proses yang diasosiasikan dengan PCB.
  2. Perubahan rancangan struktur dan sernantiks PCB dapat berdampak ke sejurnlah modul sistern operasi yang memakai PCB.

Pengaksesan Informasi pada PCB [2]

- Kedua masalah tersebut memberi gagasan agar sernua rutin sistern operasi melewati satu rutin khusus yaitu rutin penanganan PCB dalam mengakses PCB. Tugas rutin adalah memproteksi PCB dan menjadi perantara pembacaan dan penulisan PCB.
  - Masalah pertarna dapat dicegah karena rutin penanganan PCB akan selalu menjaga agar PCB tidak rusak.
  - Masalah keduajelas langsung teratasi karena antarrnuka terhadap, rutin-rutin lain masih tetap dipertahankan walau rincian-rincian PCB diubah. Rutin-rutin sistern operasi yang memakai antarmuka tidak perlu diubah.
- Teknik ini menghendaki didefinisikan antarmuka rutin penanganan PCB dan rutin-rutin lain dengan baik. Kelemahan teknik ini adalah adanya *overhead* kinerja karena harus memanggil rutin penanganan PCB. Pengaksesan langsung terhadap, PCB tentu lebih cepat daripada harus memanggil rutin penanganan PCB.

Tahap Penciptaan Proses [1]

- Penciptaan proses dapat disebabkan beragam sebab. Penciptaan proses ini meliputi beberapa tahap. Tahap-tahap penciptaan proses adalah sebagai berikut:
  1. Beri satu identifier unik ke proses baru. Isian baru ditambahkan ke tabel proses utama yang berisi satu isian per proses.
  2. Alokasikan ruang untuk proses.
  3. PCB harus diinisalisasi.
  4. Kaitan-kaitan, antar tabel dan senarai yang cocok dibuat.
  5. Bila diperlukan struktur data lain maka segera buat struktur data itu

Pengalihan Proses

- Kelihatannya pengalihan proses *(process switching)* adalah sepele. Pada suatu saat, proses *running* diinterupsi dan sistem operasi memberi proses lain *state running* dan menggilir kendali ke proses itu.
- Dalam hal ini muncul beberapa masalah, yaitu:
  1. Kejadian-kejadian apa yang memicu alih proses?
  2. Masalah lain adalah terdapatnya perbedaan antara alih proses *(process -switching)* dan alih konteks *(context- switching).*
  3. Apa yang harus dilakukan sistem operasi terhadap beragam struktur data yang dibawah kendalinya dalam alih proses?

Kejadian Penyebab Pengalihan Proses

- Kejadian-kejadian yang menyebabkan terjadinya alih proses adalah:
  1. Interupsi sistem.
  2. Trap.
  3. *Supervisor call.*

Interupsi Sistem

- Interupsi sistem disebabkan kejadian eksternal dan tak bergantung proses yang saat itu sedang *running.*
- **Contoh :** Selesainya operasi masukan/keluaran.
- Pada kejadian interupsi, kendali lebih dulu ditransfer ke *interrupt handler* yang melakukan penyimpanan data-data dan kemudian beralih ke rutin sistem operasi yang berkaitan dengan tipe interupsi itu.
- Tipe-tipe interupsi antara lain:
  - Interupsi *clock (clock interrupt).*
  - Interupsi masukanlkeluaran (110 *interrupt).*
  - *Page/memory fault.*

Interupsi Clock

- Sistem operasi (penjadwal) menentukan apakah proses yang sedang *running* telah mengeksekusi selamajatah waktunya. Jika telah mencapaijatahnya maka proses dialihkan ke *state ready* dan proses lain dijadwalkan *running.*

Interupsi I/O [1]

- Kejadian dimana peralatan masukan/keluaran melakukan interupsi meminta layanan sistem operasi. Sistem operasi segera menentukan aksi-aksi masukanlkeluaran yang harus dilakukan.

- Pemroses menemui pengacuan alamat memori maya yang tidak terdapat di memori utarna (fisik). Sistem operasi segera memerintahkan untuk mengambil page yang terdapat alamat yang dimaksud untuk dipindah ke memori utama.

Interupsi I/O [2]

- Trap adalah interupsi karena terjadinya kesalahan atau kondisi kekecualian *(exception conditions)* yang dihasilkan proses yang *running,* seperti usaha *illegal* dalam mengakses file.
- Dengan adanya trap, sistem operasi menentukan apakah kesalahan yang dibuat merupakan kesalahan fatal?
  - Jika fatal, proses yang saat itu running disingkirkan dan terjadi alih proses.
  - Jika kesalahan tidak fatal maka bergantung sifat kesalahan dan rancangan sistern operasi. Kemungkinan yang dilakukan adalah menjalankan prosedur pemulihan atau memperingatkan ke pemakai.
- Saat terjadi trap, mungkin terjadi pengalihan proses mungkin pula *resume proses*.

Interupsi I/O [3]

- Supervisor *call* yaitu panggilan meminta atau mengaktifkan bagian sistem operasi.
- **Contoh :** Proses pemakai running meminta layanan masukanlkeluaran seperti membuka file. Panggilan ini menghasilkan transfer ke rutin bagian sistem operasi. Biasanya, penggunaan system *call* membuat proses pemakai *blocked k*arena diaktifkan proses *kernel* (sistem, operasi).

Pengalihan Proses & Pengalihan Konteks

- Banyak buku teks sistem operasi menyamakan antara pengalihan proses (process *switching)* dan pengalihan konteks *(context-switching).* Tidak terdapat istilah untuk aksi penanganan interupsi. Kita membedakan antara istilah pengalihan proses dan pengalihan konteks.

Pengalihan Konteks [1]

- Pengalihan konteks dapat terj adi tanpa pengalihan state proses yang sedang running, sedang pengalihan proses pasti melibatkan juga pengalihan konteks.
- **Siklus penanganan interupsi adalah**
  1. **Pemroses menyimpan konteks program saat itu yang sedang dieksekusi ke *stack.***
  2. **Pemroses men-set register PC dengan alamat awal program untuk *interrupt handler*.**
- Setelah kedua aktivitas itu, pemroses melanjutkan menjalankan instruksi-instruksi berikutnya *& interrupt handler* yang melayani interupsi. Pelaksanaan interupsi ini belum tentu niengakibatkan pengalihan ke proses lain [yaitu pengalihan PCB proses dari senarai *running* ke senarai lain *(blocked, ready,* dan sebagainya), dan sebaliknya.]. Kita menyebut pengalihan konteks adalah untuk pengalihan sementara yang singkat, misalnya untuk mengeksekusi program *interrupt handler.*
- Setelah selesai penanganan interupsi maka konteks yang terdapat pada stack dikembalikan sehingga kembali ke konteks proses semula tanpa terjadi pengalihan ke proses lain.

Pengalihan Konteks [2]

- Pengalihan proses adalah terjadi jika proses yang *running* beralih menjadi *state* lain *(ready, blocked,* dan sebagainya), kemudian sistern operasi harus membuat perubahan-perubahan berarti terhadap lingkungannya. Rincian-rincian dalam pelaksanaan pengalihan proses di bahas setelah ini.

Pengalihan Konteks [3]

- Pengalihan proses terjadijika proses yang *running* beralih menjadi *state* lain *(ready, blocked,* dan sebagainya) kemudian sistem. operasi membuat perubahan-perubahan berarti terhadap lingkungan.
- Langkah-langkah yang terlibat dalam pengalihan proses sebagai berikut:
  1. Simpan konteks pernroses, termasuk *register* PC dan *register- register* lain.
  2. Perbarui PCB proses yang *running.* Pelaksanaan termasuk mengubah *state* proses menjadi salah satu *state (ready, blocked, suspendedready,* dan sebagainya). *Field-field* yang relevanjuga diperbarui misaInya alasan meninggalkan *state runningdan i*nformasi akunting.
  3. Pindahkan PCB proses ke senarai yang cocok *(ready, blocked,* dan sebagainya).
  4. Pilih satu proses lain untuk dieksekusi sesuai dengan teknik penjadwalan.
  5. Perbarui PCB proses yang dipilih termasuk perubahan *state* menjadi *running.*
  6. Perbarui struktur-struktur data manaJemen memori. Pekerjaan ini sesuai dengan pengefolaan translasi alamat.
  7. Kembalikan konteks pernroses dengan konteks simpanan yang memberitahu konteks proses terakhir saat dialihkan dari *state running,* Pengembalian konteks ini dilakukan dengan memuatkan nilai-nilai *registerPC* dan *register-register* lain dengan nilai konteks yang tersimpan.
- Pengalihan proses melibatkan pengalihan konteks dan perubahan *state,* memerlukan usaha lebih besar daripada pengalihan konteks.