



MANAJEMEN PROSES dan INTERUPSI

Mata Kuliah Sistem Operasi
Akmal, S.Si, MT



Proses

- Proses adalah abstraksi tentang suatu program yang sedang dijalankan
- Sistem operasi bertanggung jawab atas aktivitas-aktivitas yang berkaitan dengan manajemen proses seperti:
 - Pembuatan dan penghapusan proses pengguna dan sistem proses.
 - Menunda atau melanjutkan proses.
 - Menyediakan mekanisme untuk proses sinkronisasi.
 - Menyediakan mekanisme untuk proses komunikasi.
 - Menyediakan mekanisme untuk penanganan deadlock.
- Komponen proses
 - program yang sedang dieksekusi
 - data
 - sumber daya (resources)
 - status dari eksekusi
- Semua komputer modern bisa melakukan beberapa pekerjaan dalam saat yang sama.



MODEL PROSES

■ Multiprogramming (Multitasking)

- Manajemen banyak proses pada satu pemroses
- Melakukan proses satu persatu secara bergantian dalam waktu yang sangat cepat / bersamaan (hardware level). Setiap proses mempunyai satu virtual CPU.

Program-program yang dijalankan bersifat:

- Saling tak bergantung (independen)
- Proses terpisah satu dari lainnya dan tidak saling berpengaruh.
- Satu program pada satu saat (*one program at any instant*)
- Pada satu waktu sesungguhnya hanya satu proses yang dilayani pemroses menggunakan interleave (saling melanjutkan/bersambung) bukan overlap diantara program-program.

■ Pseudoparallelism

Melakukan lebih dari satu pekerjaan dalam waktu yang bersamaan / pseudoparallelism (user level).

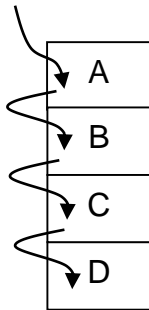
Semua S/W dikelompokkan menjadi sejumlah proses sekuensial.

Menggunakan model pseudoparalelisme

Suatu prosesor tunggal dapat dibagi diantara beberapa proses menggunakan suatu algoritma penjadwalan.

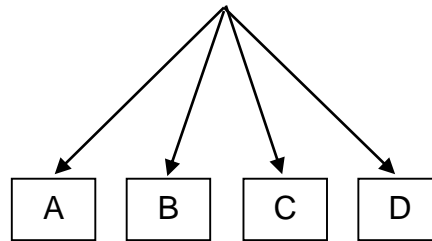
Secara konsep setiap proses mempunyai 1 virtual CPU, tetapi pada kenyataannya adalah multiprogramming. Maka akan lebih mudah ***menganggap*** kumpulan proses yang berjalan secara parallel.

Satu
program counter



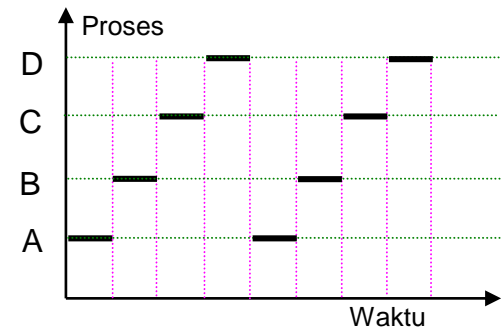
(a)

Empat
program counter



(b)

Model
Proses



(c)

Keterangan :

- a. multiprogramming untuk 4 program di memori
- b. model konseptual untuk 4 proses independent, sequential
- c. hanya 1 program yang aktif dalam 1 waktu = pseudoparalel

Contoh :

computer scientist (cs) sedang memasak kue (resep=program (algoritma), cs=prosesor, bahan=input data, proses = baca resep, campur bahan dan panggang kue)

Anaknya menangis digigit lebah, cs berhenti dan memberi pertolongan (prioritas lebih tinggi), selesai lanjutkan dengan masak lagi



ISTILAH LAIN

■ **Multiprocessing**

- Manajemen banyak proses di komputer multiprocessor (banyak pemroses didalamnya)
- Dulunya sistem ini hanya terdapat disistem besar, mainframe dan mini komputer.
- Saat ini komputer workstation telah dapat dilengkapi multiprocessor.
- Sistem operasi Microsoft Windows NT, Vista, Windows 7/8, UNIX, Linux menyediakan dukungan multiprocessing.

■ **Distributed processing**

- Manajemen banyak proses yang dieksekusi di banyak sistem komputer yang tersebar (terdistribusi)



HIRARKI PROSES

- Ada cara untuk menghapus dan membentuk proses
- Pada UNIX , proses dibentuk dengan (system call) FORK
- Proses utama dan subprocess yang dipanggil dapat berjalan secara paralel
- Pada MS-DOS, system call me-load suatu file biner tertentu ke memori dan mengeksekusinya sebagai subprocess.
- Proses utama dan subprocess tidak dapat berjalan secara paralel



KEADAAN PROSES

- Proses butuh berinteraksi dengan proses lain dan perlu diketahui kondisi dari proses lainnya itu.

Ada 3 keadaan proses :

1. Running

- Kondisi dimana suatu proses sedang menggunakan CPU pada saat tersebut

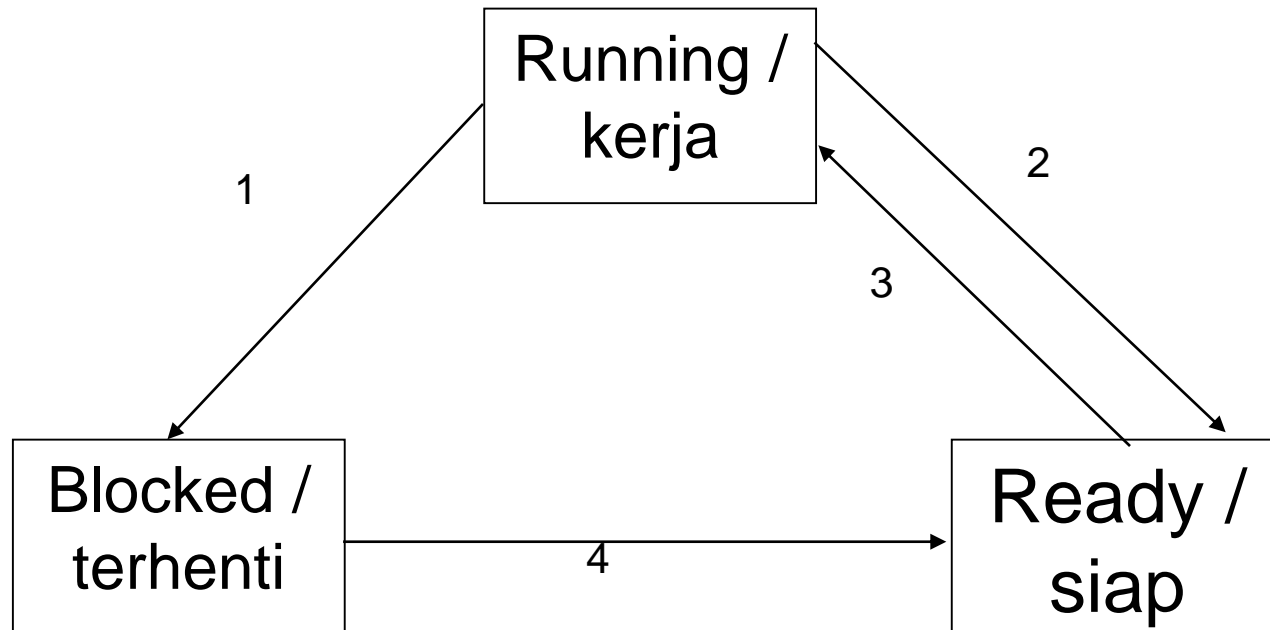
2. Ready

- Kondisi dimana proses sebenarnya runnable, namun berhenti sementara karena CPU sedang digunakan proses lain

3. Blocked

- Kondisi dimana proses belum dapat dijalankan karena menunggu kelengkapan proses eksternal

4 Transisi Status Proses





4 Transisi Proses

1. Running menjadi blocked

- transisi ini terjadi karena ditemukan proses tidak dapat dilanjutkan karena ada sesuatu yang terjadi

2. Running menjadi ready

- terjadi karena penjadwal proses memutuskan bahwa suatu proses terlalu lama menggunakan CPU

3. Ready menjadi running

- terjadi jika semua proses telah mendapatkan bagiannya menggunakan CPU dan saatnya proses dalam kondisi ready untuk menggunakan CPU

4. Blocked menjadi ready


- terjadi jika kelengkapan eksternal yang ditunggu proses sudah terpenuhi



IMPLEMENTASI PROSES

- OS menggunakan tabel proses, satu entry per proses
- Tabel proses berisi :
 - informasi tentang status proses
 - program counter
 - stack pointer
 - alokasi memori
 - status-status file yang sedang dibuka
 - informasi akunting dan penjadwalan
 - segala sesuatu yang perlu disimpan pada saat berpindah dari kondisi running ke ready
- Isi tabel proses secara umum memuat :
manajemen proses, manajemen memori, manajemen file.

Contoh Tabel Proses :



Process management	Memory management	File management
Register	Pointer to text segment	UMASK mask
Program counter	Pointer to data segment	Root directoy
Program status word	Pointer to bss segment	Working directory
Stack pointer	Exit status	File descriptions
Process state	Signal status	Effective uid
Time when process started	Process id	Effective gid
CPU time used	Parent process	System call parameters
Children's CPU time	Process group	Various flag bits
Time of next alarm	Real uid	
Message queue pointers	Effective uid	
Pending signal bits	Real gid	
Process id	Effective gid	
Various flag bits	Bit maps for signals	
	Various flag bits	



Interupsi

- **Interupsi** : Kerja prosesor pada suatu proses terhenti oleh pensaklaran konteks.
- **Pensaklaran konteks** : perubahan kegiatan prosesor dari proses ke proses yang terjadi diantara proses sistem / proses aplikasi
- **Konteks** : kegiatan prosesor terhadap sesuatu hal, berasal dari sistem operasi, sistem bahasa dan sistem utilitas.
- **Blok kendali proses** : suatu bagian memori untuk mencatat keadaan proses, yang terbagi atas wilayah dimana setiap wilayah untuk mencatat informasi yang berbeda.



2 Cara Interupsi pada Processor

- *Interupsi langsung*
Berasal dari luar prosesor (peripheral / alat mengirim sinyal kepada prosesor untuk meminta pelayanan)
- *Interupsi Tanya / Polling*
Berasal dari prosesor (prosesor secara bergiliran mengecek apakah ada peripheral yang memerlukan pelayanan atau tidak)

Interupsi dapat di-enable dan disable tergantung pada levelnya.



Pembangkit interupsi dapat berasal dari :(1)

- ***Program,***
di dalam program telah dirancang pada bagian tertentu akan terjadi pensaklaran konteks, yang menimbulkan interupsi, contohnya pada saat penggunaan alat / prosesor secara bergantian.
- ***Prosesor,***
prosesor sendiri dapat membangkitkan interupsi, yang biasa mengolah logika dan aritmatika. Jika melampoi ukuran tampung register di dalam prosesor, maka terjadi kekeliruan yang akan menginterupsi kerjanya sendiri dan menyerahkan kendali prosesor pada sistem operasi. Misalnya pembagian dengan bilangan nol.
- ***Satuan kendali,***
tugas untuk melaksanakan interupsi terletak pada satuan kendali, sehingga satuan kendali dapat membangkitkan interupsi. Misalnya kekeliruan instruksi
- ***Kunci waktu / clock,***
menggunakan interupsi berkala. Misalnya pada program looping yang tak terhingga, diinterupsi pada setiap selang waktu 60 detik.



Pembangkit interupsi dapat berasal dari :(2)

- ***Peripheral I/O,***

I/O jika akan bekerja memberitahukan pada prosesor dengan interupsi prosesor dan juga ketika pekerjaan selesai atau pada saat terjadi kekeliruan paritas.

- ***Memori,***

karena terjadi kekeliruan, misalnya ketika prosesor ingin mencapai alamat memori yang terletak di luar bentangan alamat memori yang ada.

- ***Sumber daya lain,***

misal dibangkitkan oleh operator sistem komputer yang mengerti cara interupsi.



Tindak lanjut interupsi :

- ***Penata interupsi / interrupt handler***

jika terjadi interupsi, maka kendali prosesor diserahkan ke bagian penata interupsi pada sistem operasi, maka penata interupsi inilah yang melaksanakan interupsi.

- Instruksi yang sedang diolah oleh prosesor dibiarkan sampai selesai program.
- Penata interupsi merekam semua informasi proses ke dalam blok kendali proses.
- Penata interupsi mengidentifikasi jenis dan asal interupsi.
- Penata interupsi mengambil tindakan sesuai dengan yang dimaksud interupsi.
- Penata interupsi mempersiapkan segala sesuatu untuk melanjutkan proses yang diinterupsi.

- ***Penata keliru / error handler***

yaitu interupsi karena kekeliruan pada pengolahan proses dan bagian pada sistem operasi yang menata kegiatan akibat kekeliruan adalah penata keliru.

- *Pemulihan*, komputer telah dilengkapi dengan sandi penemuan dan pemulihan kekeliruan, contohnya telah dilengkapi dengan sandi Hamming sehingga ketika menemukan kekeliruan sandi akan mengoreksi kekeliruan itu, proses pulih ke bentuk semula sebelum terjadi kekeliruan.
- *Pengulangan*, mengatur agar proses yang membangkitkan interupsi keliru dikerjakan ulang, jika kekeliruan dapat diatasi maka proses akan berlangsung seperti biasa, jika tidak teratasi maka interupsi akan menempuh tindak lanjut keluar dari proses.
- *Keluar dari proses*, penata keliru menyiapkan tampilan berita keliru dari monitor, setelah itu prosesor keluar dari proses, ini adalah tindakan terakhir jika tidak dapat menolong proses yang keliru tersebut.