

Sistem Operasi

Pertemuan 05

Alauddin Maulana Hirzan, S.Kom., M.Kom.
NIDN. 0607069401

Fakultas Teknologi Informasi dan Komunikasi, Universitas Semarang



- 1 Manajemen Proses
- 2 Proses & Thread
- 3 Siklus Hidup Proses
- 4 Sinkronisasi Proses
- 5 Inter-Process Communication

Manajemen Proses

Definisi Proses

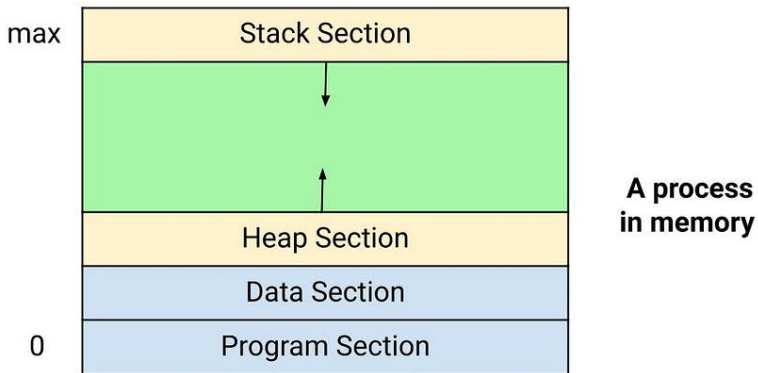
Definisi:

Dalam komputasi, suatu proses dapat didefinisikan sebagai turunan dari program komputer yang sedang dijalankan. Hal ini mencakup kode program, aktivitasnya saat ini, dan sumber daya sistem terkait, seperti memori, waktu CPU, register, dan perangkat I/O.

Suatu proses adalah entitas dinamis; dia mengalami transisi status, dari baru ke berjalan, menunggu, siap, dan dihentikan, saat dijalankan dalam sistem operasi.

Manajemen Proses

Definisi Proses



Manajemen Proses

Definisi Manajemen Proses

Definisi Manajemen Proses:

Manajemen proses mengacu pada administrasi proses dalam sistem operasi. Ini melibatkan pembuatan, penjadwalan, penghentian, dan sinkronisasi proses untuk memastikan pelaksanaan tugas yang efisien dan teratur.

Manajemen proses menyediakan mekanisme untuk mengalokasikan sumber daya sistem, seperti waktu CPU dan memori, ke proses dengan cara yang adil dan terkendali. Selain itu, ia mengawasi komunikasi dan sinkronisasi antar-proses untuk memfasilitasi kolaborasi antar proses.

Manajemen Proses

Pentingnya Manajemen Proses

Mengelola proses sangat penting karena beberapa alasan:

- **Pemanfaatan Sumber Daya:** Manajemen proses yang efektif memastikan pemanfaatan sumber daya sistem secara optimal.
- **Konkurensi:** Dalam sistem operasi modern, banyak proses sering kali berjalan secara bersamaan.
- **Isolasi dan Perlindungan:** Proses memberikan tingkat isolasi satu sama lain, mencegah mereka mengganggu ruang memori satu sama lain atau menyebabkan ketidakstabilan sistem.

Manajemen Proses

Pentingnya Manajemen Proses

- **Toleransi Kesalahan:** Manajemen proses berperan dalam menjaga stabilitas sistem dan memulihkan dari kegagalan. Ini memonitor proses untuk kesalahan, memfasilitasi penanganan kesalahan, dan, dalam beberapa kasus, mendukung mekanisme pemulihan proses.
- **Pengalaman Pengguna:** Manajemen proses yang efisien berkontribusi pada lingkungan komputasi yang responsif dan andal, meningkatkan pengalaman pengguna dengan meminimalkan penundaan, mencegah kerusakan sistem, dan mengoptimalkan kinerja sistem.

Manajemen Proses

Peran Manajemen Proses untuk Alokasi

Manajemen proses sangat penting dalam alokasi sumber daya dan multitasking dalam sistem operasi:

- **Alokasi Sumber Daya:** Manajemen proses mengalokasikan sumber daya sistem, seperti waktu CPU, memori, dan perangkat I/O, di antara proses-proses yang bersaing berdasarkan prioritas, kebijakan penjadwalan, dan ketersediaan sumber daya.
- **Multitasking:** Multitasking melibatkan eksekusi beberapa proses secara bersamaan, memungkinkan pengguna menjalankan banyak aplikasi secara bersamaan. Manajemen proses menjadwalkan dan mengoordinasikan proses-proses ini, beralih di antara proses-proses tersebut dengan cepat untuk memberikan ilusi eksekusi paralel.

- 1 Manajemen Proses
- 2 Proses & Thread**
- 3 Siklus Hidup Proses
- 4 Sinkronisasi Proses
- 5 Inter-Process Communication

Proses & Thread

Definisi Proses & Thread

Definisi Proses:

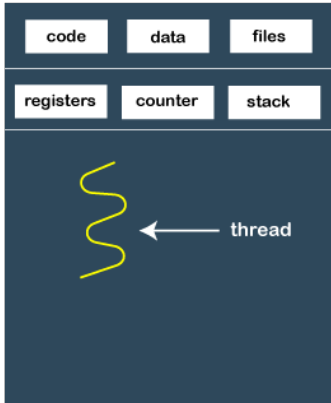
Suatu proses dapat didefinisikan sebagai program yang sedang dieksekusi. Ini adalah contoh dari program komputer yang sedang dijalankan.

Definisi Thread:

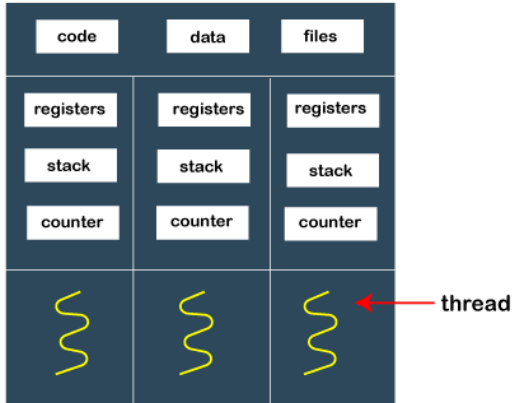
Sebaliknya, sebuah thread dapat dianggap sebagai proses yang ringan. Ini adalah unit eksekusi terkecil dalam suatu proses. Thread berbagi ruang memori dan sumber daya sistem yang sama dengan proses yang menjadi miliknya. Beberapa thread dalam satu proses dapat dijalankan secara bersamaan, memungkinkan eksekusi tugas secara paralel dalam proses yang sama.

Proses & Thread

Definisi Proses & Thread



Single-threaded process



Multi-threaded process

Proses & Thread

Kunci Perbedaan Proses & Thread

- **Alokasi sumber daya:**
 - ① Proses memiliki ruang alamat dan sumber daya sistemnya sendiri yang terpisah.
 - ② Thread dalam proses yang sama berbagi ruang alamat dan sumber daya yang sama.
- **Ketergantungan:**
 - ① Proses-prosesnya independen satu sama lain dan beroperasi secara terpisah.
 - ② Thread dalam proses yang sama berbagi sumber daya dan dapat berkomunikasi secara langsung satu sama lain.

Proses & Thread

Kunci Perbedaan Proses & Thread

- **Kelebihan Beban:**
 - ➊ Membuat suatu proses biasanya memerlukan lebih banyak overhead, karena melibatkan duplikasi seluruh struktur proses.
 - ➋ Thread lebih ringan bobotnya dan memiliki overhead yang lebih sedikit saat dibuat, karena thread berbagi sumber daya dalam proses.
- **Kontrol:**
 - ➊ Proses dikelola secara independen oleh sistem operasi.
 - ➋ Thread dikelola dalam konteks proses dimana thread tersebut berada, dan penjadwalan thread ditangani oleh penjadwal thread proses.

Proses & Thread

Kunci Perbedaan Proses & Thread

- **Isolasi kesalahan:**
 - 1 Dalam suatu proses, jika salah satu proses mengalami crash, maka tidak mempengaruhi proses lainnya.
 - 2 Di thread, masalah pada satu thread berpotensi mempengaruhi keseluruhan proses, menyebabkan ketidakstabilan.

Proses & Thread

Kelebihan Menggunakan Thread

Kelebihan:

- **Responsiveness:** Thread dapat membuat aplikasi lebih responsif dengan memungkinkan eksekusi tugas secara bersamaan.
- **Berbagi Sumber Daya:** Thread dalam proses yang sama dapat berbagi sumber daya dengan lebih efisien dibandingkan dengan proses terpisah, sehingga menghasilkan pemanfaatan sumber daya yang lebih baik.
- **Efisiensi:** Membuat dan mengelola thread biasanya memerlukan overhead yang lebih sedikit dibandingkan dengan proses, karena thread berbagi ruang memori dan sumber daya yang sama dengan proses induk.

Proses & Thread

Kekurangan Menggunakan Thread

Kekurangan:

- **Kompleksitas:** Pemrograman multithread menimbulkan kompleksitas seperti kondisi balapan, kebuntuan, dan masalah sinkronisasi, yang mungkin sulit untuk di-debug dan dipelihara.
- **Kesulitan dalam Debugging:** Men-debug aplikasi multithread bisa lebih menantang dibandingkan dengan aplikasi single-thread karena sifat eksekusi thread yang non-deterministik.
- **Perselisihan Sumber Daya:** Thread dalam proses yang sama mungkin bersaing untuk mendapatkan sumber daya bersama, sehingga berpotensi menyebabkan penurunan kinerja jika tidak dikelola dengan benar.

- 1 Manajemen Proses
- 2 Proses & Thread
- 3 Siklus Hidup Proses**
- 4 Sinkronisasi Proses
- 5 Inter-Process Communication

Siklus Hidup Proses

Cara Manajemen Proses

Sistem operasi (OS) bertanggung jawab untuk mengelola berbagai proses yang berjalan pada sistem komputer. Proses adalah turunan dari program yang sedang berjalan, termasuk kode program, data, dan sumber daya yang digunakannya. Berikut cara OS mengelola proses:

- 1 Pembuatan Proses
- 2 Penjadwalan Proses
- 3 Eksekusi Proses
- 4 Komunikasi Proses
- 5 Terminasi Proses

Siklus Hidup Proses

Cara Manajemen Proses

1. Pembuatan Proses

Saat pengguna atau sistem memulai suatu program, OS membuat proses baru untuk program tersebut. Hal ini melibatkan pengalokasian memori, pembuatan blok kendali proses (PCB), dan inisialisasi struktur data yang diperlukan.

2. Penjadwalan Proses

OS memutuskan proses mana yang akan dieksekusi selanjutnya menggunakan algoritma penjadwalan. Hal ini melibatkan pertimbangan faktor-faktor seperti prioritas, ketersediaan sumber daya, dan keadilan.

Siklus Hidup Proses

Cara Manajemen Proses

3. Eksekusi Proses:

Setelah dijadwalkan, OS mengalokasikan waktu CPU untuk proses, memungkinkannya menjalankan instruksinya.

4. Komunikasi Proses:

Proses mungkin perlu berkomunikasi satu sama lain, baik melalui memori bersama, penyampaian pesan, atau mekanisme komunikasi antar proses lainnya.

5. Penghentian Proses:

Ketika suatu proses menyelesaikan eksekusinya, secara sukarela atau karena kesalahan, OS melepaskan sumber daya yang dialokasikan untuk proses itu.

Siklus Hidup Proses

Status Proses

Manajemen proses tidak akan terlepas dari status dari proses itu sendiri, terdapat berbagai macam status proses seperti berikut:

1. **Baru / New:** Proses sedang dibuat. Pada tahap ini, OS menginisialisasi blok kendali proses (PCB) dan mengalokasikan sumber daya.
2. **Siap / Ready:** Proses siap dieksekusi tetapi menunggu CPU. Ia memiliki semua sumber daya yang dibutuhkan untuk menjalankannya tetapi menunggu gilirannya dijadwalkan.
3. **Berjalan / Running:** CPU secara aktif menjalankan instruksi untuk proses tersebut. Dalam lingkungan multitasking, beberapa proses mungkin berada dalam keadaan ini, dengan CPU berpindah di antara proses tersebut menggunakan algoritma penjadwalan.

Siklus Hidup Proses

Status Proses

4. **Diblokir / Waiting:** Proses tidak dapat dijalankan sampai suatu peristiwa terjadi. Ini mungkin menunggu input pengguna, operasi I/O selesai, atau sinyal dari proses lain.
5. **Dihentikan / Stopped :** Proses telah selesai dijalankan dan dihentikan secara sukarela atau dihentikan karena kesalahan. OS melepaskan semua sumber daya yang terkait dengan proses tersebut.

Siklus Hidup Proses

Transisi Status Proses

Status Proses tersebut pastinya akan melalui transisi satu sama lainnya. Berikut ini adalah Tahapan Transisi Status Proses:

1. **Penciptaan/Creation:** Transisi dari status "Baru" ke status "Siap".
2. **Pengiriman/Dispatch:** Transisi dari status "Siap" ke status "Berjalan".
3. **Pemblokiran/Blocking:** Transisi dari status "Berjalan" ke status "Diblokir".
4. **Bangun/Wake-up:** Transisi dari status "Diblokir" kembali ke status "Siap".
5. **Penyelesaian/Completion:** Transisi dari keadaan apa pun ke keadaan "Dihentikan".

Siklus Hidup Proses

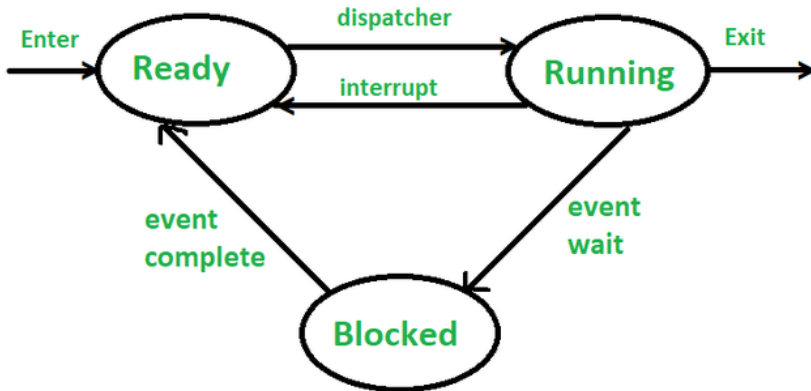
Model Manajemen Proses

Sistem Operasi memiliki beberapa model manajemen proses seperti berikut:

- 1 **Three-States Model**
- 2 **Five-States Model**
- 3 **Six-States Model**

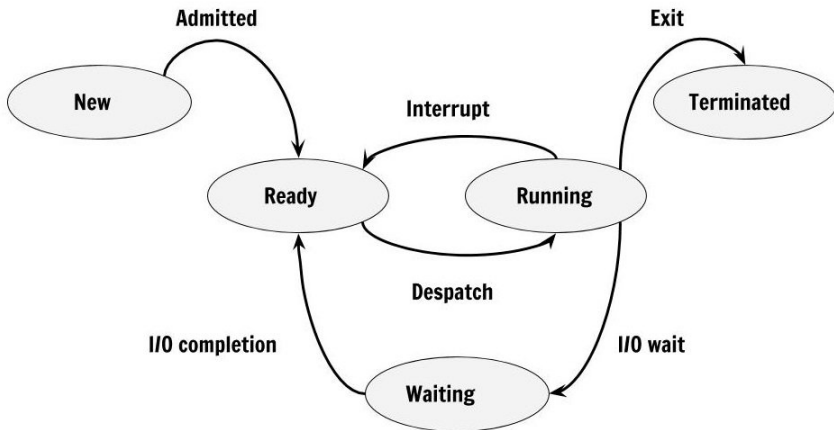
Siklus Hidup Proses

Model Manajemen Proses - Three-States Model



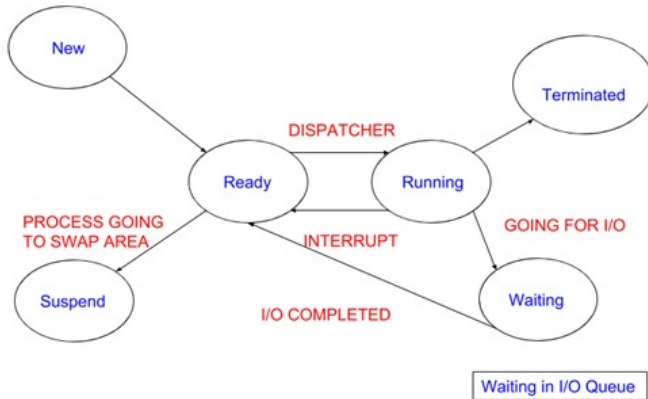
Siklus Hidup Proses

Model Manajemen Proses - Five-States Model



Siklus Hidup Proses

Model Manajemen Proses - Six-States Model



- 1 Manajemen Proses
- 2 Proses & Thread
- 3 Siklus Hidup Proses
- 4 Sinkronisasi Proses**
- 5 Inter-Process Communication

Sinkronisasi Proses

Definisi Sinkronisasi Proses

Sinkronisasi proses mengacu pada koordinasi proses bersamaan dalam lingkungan multitasking atau multiproses untuk memastikan bahwa proses tersebut dijalankan dengan benar tanpa mengganggu satu sama lain.

Dalam sistem multitasking, beberapa proses atau thread mungkin berjalan secara bersamaan, berbagi sumber daya seperti memori, file, atau perangkat keras. Mekanisme sinkronisasi proses digunakan untuk mengontrol urutan eksekusi dan akses ke sumber daya bersama untuk mencegah masalah seperti inkonsistensi data, kondisi balapan, dan kebuntuan.

Sinkronisasi Proses

Alasan Sinkronisasi Proses

Alasan menggunakan Sinkronisasi:

- **Sumber Daya Bersama:**

Dalam lingkungan multitasking, proses sering kali berbagi sumber daya seperti memori, file, dan perangkat. Tanpa sinkronisasi yang tepat, akses bersamaan ke sumber daya bersama dapat menyebabkan kerusakan dan inkonsistensi data.

- **Kondisi Balapan:**

Kondisi balapan terjadi ketika beberapa proses mencoba mengubah data bersama secara bersamaan, sehingga menyebabkan perilaku yang tidak dapat diprediksi. Mekanisme sinkronisasi membantu mencegah kondisi balapan dengan menerapkan saling pengecualian.

Sinkronisasi Proses

Alasan Sinkronisasi Proses

- **Kebuntuan:**

Deadlock terjadi ketika dua atau lebih proses tidak dapat dilanjutkan karena masing-masing proses menunggu proses lainnya melepaskan sumber daya. Teknik sinkronisasi membantu mencegah kebuntuan dengan menyediakan metode alokasi dan pelepasan sumber daya.

- **Kebenaran dan Keandalan:**

Sinkronisasi memastikan kebenaran dan keandalan program bersamaan dengan menerapkan urutan eksekusi dan akses ke sumber daya bersama yang konsisten.

Sinkronisasi Proses

Metode Sinkronisasi Proses

Terdapat empat metode untuk melakukan sinkronisasi proses:

1. **Mutex (Mutual Exclusion):**

Semafor biner hanya mengizinkan satu proses pada satu waktu untuk mengakses sumber daya bersama

2. **Semafor:**

Variabel integer yang digunakan untuk memberi sinyal antar proses.

Sinkronisasi Proses

Metode Sinkronisasi Proses

3. Monitor:

Konstruksi sinkronisasi tingkat tinggi yang merangkum data bersama dan prosedur untuk mengaksesnya.

4. Variabel Bersyarat:

Digunakan bersama dengan mutex atau monitor untuk menunggu kondisi tertentu.

- 1 Manajemen Proses
- 2 Proses & Thread
- 3 Siklus Hidup Proses
- 4 Sinkronisasi Proses
- 5 Inter-Process Communication**

Inter-Process Communication

Definisi Inter-Process Communication

Definisi:

Komunikasi Antar-Proses memungkinkan proses untuk menyinkronkan tindakan mereka dan bertukar data. Dalam lingkungan multi-proses, proses mungkin perlu bekerja sama untuk menyelesaikan tugas atau berbagi informasi. IPC memfasilitasi kolaborasi ini dengan menyediakan metode bagi proses untuk mengirim dan menerima pesan, berbagi memori, atau menyinkronkan eksekusinya.

Inter-Process Communication

Metode Inter-Process Communication

Metode-metode dari IPC:

1. Pipa:

Pipa menyediakan saluran komunikasi satu arah antara sepasang proses.

2. Antrean Pesan:

Antrian pesan memungkinkan proses untuk berkomunikasi dengan mengirimkan pesan ke antrian.

3. Berbagi memori:

Memori bersama memungkinkan proses untuk berbagi wilayah memori yang dapat diakses oleh banyak proses.

Inter-Process Communication

Metode Inter-Process Communication

Metode-metode dari IPC:

4. Soket:

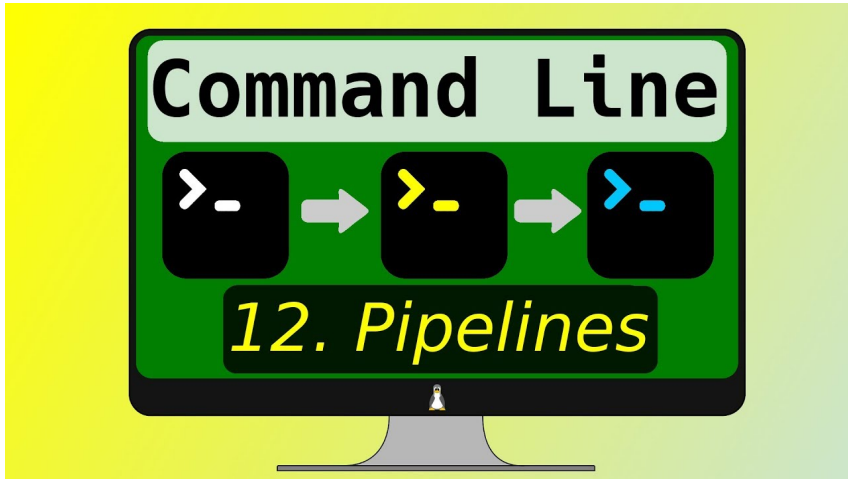
Soket menyediakan titik akhir komunikasi yang memungkinkan proses berkomunikasi melalui jaringan atau antar proses pada sistem yang sama.

5. Sinyal:

Sinyal adalah interupsi perangkat lunak yang memungkinkan proses saling memberitahukan peristiwa atau menangani kondisi luar biasa. tertentu.

Inter-Process Communication

Metode Inter-Process Communication



Terima Kasih