

# **MAKALAH SISTEM OPERASI**

Disusun Untuk Memenuhi Salah Satu Tugas Mata Kuliah Sistem Operasi



Disusun oleh :

Melina Amelia

2206152

Informatika -E

**PROGRAM STUDI TEKNIK INFORMATIKA**

**INSTITUT TEKNOLOGI GARUT**

**2023**

## **KATA PENGANTAR**

Puji syukur dipanjatkan ke hadirat Allah SWT. atas segala rahmat-Nya sehingga makalah ini dapat diselesaikan. Tidak lupa Saya ucapkan terima kasih atas bantuan pihak-pihak yang telah menyumbangkan baik ide maupun materi.

Saya sangat berharap Makalah ini dapat memberikan pengetahuan dan pengalaman bagi para pembaca. Bahkan Saya berharap lebih banyak lagi agar pembaca dapat mempraktekkan Makalah ini dalam kehidupan sehari-hari.

Saya selaku penulis merasa bahwa karena keterbatasan pengetahuan dan pengalaman, masih banyak kekurangan dalam penyusunan Makalah ini. Oleh karena itu, Saya sangat mengharapkan kritik dan saran yang membangun dari para pembaca demi kesempurnaan Makalah ini (Urwatul, 2022).

Garut, 5 April 2023

Penulis

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

Sistem operasi adalah program yang mengontrol dan mengatur penggunaan sumber daya komputer dan menjembatani antara pengguna dengan perangkat keras komputer, dimana sistem operasi memiliki banyak komponen yang memegang peranan sangat penting. Salah satu komponennya adalah proses dan utas (multithreading).

Hal inilah yang melatarbelakangi dilakukannya pekerjaan ini yaitu memeriksa isi proses dan thread (multi-threaded). Tujuannya agar mahasiswa kurang lebih memahami apa itu proses dan thread dalam sebuah sistem operasi.

### **1.2. Rumusan Masalah**

Mengacu pada latar belakang masalah yang telah dikemukakan, maka rumusan masalah pada Makalah ini adalah :

- 1) Apa yang dimaksud dengan Proses?
- 2) Jelaskan Pengelolaan dan Status Proses?
- 3) Jelaskan apa yang dimaksud Process Control Block (PCB)?
- 4) Apa yang dimaksud dengan Thread?

### **1.3. Tujuan Penelitian**

Dengan penulisan makalah ini, penyusun mempunyai tujuan agar dapat dicapai, yaitu :

- 1) Mengetahui Definisi dari Proses
- 2) Mengetahui Pengelolaan dan Status Proses
- 3) Mengetahui Penjelasan tentang Process Control Block (PCB)
- 4) Mengetahui definisi dari Thread

## **BAB II**

### **PEMBAHASAN**

#### **2.1. Definisi Proses**

Terdapat beberapa definisi mengenai proses, antara lain :

- Merupakan konsep pokok dalam sistem operasi, sehingga masalah manajemen proses adalah masalah utama dalam perancangan sistem operasi.
- Proses adalah program yang sedang dieksekusi.
- Proses adalah unit kerja terkecil yang secara individu memiliki sumber daya dan diadalkan oleh sistem operasi. Peran sistem operasi dalam kegiatan proses adalah mengelola semua proses di sistem dan mengalokasikan sumber daya ke proses tersebut. banyak proses yang dijalankan bersamaan, dimana setiap proses mendapat bagian memori dan kendali sendiri-sendiri, sehingga setiap proses memiliki prinsip :
  - Independent, artinya program-program tersebut berdiri sendiri, terpisah dan saling tidak bergantung.
  - One program at any instant, artinya hanya terdapat satu proses yang dilayani pemroses pada satu saat. dalam multiprogramming, teknik penanganan proses adalah dengan mengeksekusi satu proses dan secara cepat beralih ke proses lainnya

#### **2.2. Pengelolaan Proses dan Status Proses**

Proses perlu dikelola karena dalam sebuah proses membutuhkan beberapa sumber daya untuk menyelesaikan tugasnya. Sumber daya tersebut dapat berupa CPU time, memori, berkas-berkas, dan perangkat-perangkat I/O.

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

Proses yang dieksekusi mempunyai lima status yang terdiri dari:

- a. new: Pembentukan suatu proses
- b. running: Instruksi-instruksi yang sedang dieksekusi

- c. waiting: Proses menunggu untuk beberapa event yang terjadi
- d. ready: Proses menunggu untuk dialirkan ke pemroses (processor)
- e. terminated: Proses telah selesai dieksekusi

### **2.3. Process Control Block (PCB)**

Setiap 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, termasuk hal-hal di bawah ini:

Status proses: status yang mungkin adalah new, ready, running, waiting, halted, dan seterusnya.

Program counter adalah suatu penghitung yang mengindikasikan alamat dari instruksi selanjutnya yang akan dieksekusi untuk proses tersebut. CPU register: Register bervariasi dalam jumlah dan tipenya, tergantung pada arsitektur komputer. Register tersebut termasuk accumulator, index register, stack pointer, general-purposes register, ditambah informasi condition-code. Bersama dengan program counter, keadaan/status informasi harus disimpan ketika gangguan terjadi, untuk memungkinkan proses tersebut berjalan/bekerja dengan benar. Informasi manajemen 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.

Informasi pencatatan: Informasi ini termasuk jumlah dari CPU dan waktu nyata yang digunakan, batas waktu, jumlah account, jumlah job atau proses, dan banyak lagi.

Informasi status I/O: Informasi termasuk daftar dari perangkat I/O yang digunakan pada proses ini, suatu daftar berkas-berkas yang sedang diakses dan banyak lagi.

PCB hanya berfungsi sebagai tempat penyimpanan informasi yang dapat bervariasi dari proses yang satu dengan yang lain

### **2.4. Definisi Thread**

Thread dalam sistem operasi dapat diartikan sebagai sekumpulan perintah (instruksi) yang dapat dilaksanakan (dieksekusi) secara sejajar dengan ulir lainnya, dengan menggunakan cara time slice (ketika satu CPU melakukan perpindahan antara satu ulir ke ulir lainnya) atau multiprocess ketika ulir-ulir tersebut dilaksanakan oleh CPU yang berbeda dalam satu sistem). Ulir sebenarnya mirip dengan proses, tapi cara berbagi sumber daya antara proses dengan ulir sangat berbeda.

Multiplethread dapat dilaksanakan secara sejajar pada sistem komputer. Secara umum multithreading melakukan time-slicing (sama dengan time-division multipleks), di mana sebuah CPU bekerja pada ulir yang berbeda, di mana suatu kasus ditangani tidak sepenuhnya secara serempak, untuk CPU tunggal pada dasarnya benar-benar melakukan sebuah pekerjaan pada satu waktu. Thread saling berbagi bagian program, bagian data dan sumber daya sistem operasi dengan thread lain yang mengacu pada proses yang sama. Thread terdiri atas thread, program counter, himpunan register, dan stack. dengan banyak kontrol thread proses dapat melakukan lebih dari satu pekerjaan pada waktu yang sama.

Thread merupakan cara dari komputer untuk menjalankan dua atau lebih task dalam waktu bersamaan, sedangkan multithreading adalah cara komputer untuk membagi-bagi pekerjaan yang dikerjakan sebagian-sebagian dengan cepat sehingga:

- Single-threading : Sebuah proses tradisional atau heavyweight process mempunyai thread tunggal yang berfungsi sebagai pengendali.
- Multi-threading : Sebuah proses dengan thread yang banyak dan mengerjakan lebih dari satu tugas pada satu satuan waktu.

## **Process Thread**

Model-model MultiThreading :

Model Many-to-One.

Model ini memetakan beberapa thread tingkatan pengguna ke sebuah thread. tingkatan kernel. Pengaturan thread dilakukan dalam ruang pengguna sehingga efisien. Hanya satu thread pengguna yang dapat mengakses thread kernel pada satu saat. Jadi Multiple thread tidak dapat berjalan secara paralel pada multiprosesor. Kekurangannya adalah ketika ada satu blocking system call, semua akan menjadi terblok juga. Contoh: Solaris.

Model One to One

Model ini memetakan setiap thread tingkatan pengguna ke setiap thread. Ia menyediakan lebih banyak concurrency dibandingkan model Many-to-One. Keuntungannya sama dengan keuntungan thread kernel. Kelemahan model ini ialah setiap pembuatan thread pengguna memerlukan tambahan thread kernel. Karena itu, jika mengimplementasikan sistem ini maka

akan menurunkan kinerja dari sebuah aplikasi sehingga biasanya jumlah thread dibatasi dalam sistem.

Contoh: Windows NT/XP/2000 , Linux, Solaris 9, OS/2.

### Model Many to Many

Model ini memultipleks banyak thread tingkatan pengguna ke thread kernel yang jumlahnya sedikit atau sama dengan tingkatan pengguna. Model ini mengizinkan developer membuat thread sebanyak yang ia mau tetapi concurrency tidak dapat diperoleh karena hanya satu thread yang dapat dijadwalkan oleh kernel pada suatu waktu. Keuntungan dari sistem ini ialah kernel thread yang bersangkutan dapat berjalan secara paralel pada multiprocessor dan lebih efisien. Contoh : Solaris 2, IRIX, HPUX

### Pustaka Thread

Pustaka Thread atau yang lebih familiar dikenal dengan Thread Library bertugas untuk menyediakan API untuk programmer dalam menciptakan dan manage thread. Ada dua cara dalam mengimplementasikan pustaka thread:

Menyediakan API dalam level pengguna tanpa dukungan dari kernel sehingga pemanggilan fungsi tidak melalui system call. Jadi, jika kita memanggil fungsi yang sudah ada di pustaka, maka akan menghasilkan pemanggilan fungsi call yang sifatnya lokal dan bukan system call. Menyediakan API di level kernel yang didukung secara langsung oleh sistem operasi. Pemanggilan fungsi call akan melibatkan system call ke kernel. Ada tiga pustaka thread yang sering digunakan saat ini, yaitu: POSIX Pthreads, Java, dan Win32. Implementasi POSIX standard dapat dengan cara user level dan kernel level, sedangkan Win32 adalah kernel level. Java API thread dapat diimplementasikan oleh Pthreads atau Win32.

### Thread Cancellation

Thread Cancellation ialah pembatalan thread sebelum tugasnya selesai. Misalnya hendak mematikan Java Virtual Machine (JVM) pada program Java. Maka sebelum JVM dimatikan seluruh thread yang berjalan harus dibatalkan terlebih dahulu. Contoh lain adalah pada masalah search. Apabila sebuah thread mencari sesuatu dalam database dan menemukan serta

mengembalikan hasilnya, thread sisanya akan dibatalkan. Thread yang akan diberhentikan biasa disebut target thread.

Pemberhentian target Thread dapat dilakukan dengan 2 cara:

Asynchronous cancellation. Suatu thread seketika itu juga membatalkan target thread.

Deferred cancellation. Suatu thread secara periodik memeriksa apakah ia harus batal, cara ini memperbolehkan target thread untuk membatalkan dirinya secara teratur.

Hal yang sulit dari pembatalan thread ini adalah ketika terjadi situasi dimana sumber daya sudah dialokasikan untuk thread yang akan dibatalkan. Selain itu kesulitan lain adalah ketika thread yang dibatalkan sedang meng-update data yang ia bagi dengan thread lain. Hal ini akan menjadi masalah yang sulit apabila digunakan asynchronous cancellation. Sistem operasi akan mengambil kembali sumber daya dari thread yang dibatalkan tetapi seringkali sistem operasi tidak mengambil kembali semua sumber daya dari thread yang dibatalkan. Alternatifnya adalah dengan menggunakan deferred cancellation. Cara kerja dari deferred cancellation adalah dengan menggunakan satu thread yang berfungsi sebagai pengindikasi bahwa target thread hendak dibatalkan. Tetapi pembatalan hanya akan terjadi jika target thread memeriksa apakah ia harus batal atau tidak. Hal ini memperbolehkan thread untuk memeriksa apakah ia harus batal pada waktu dimana ia dapat dibatalkan secara aman yang aman. Pthread merujuk sebagai cancellation points. Pada umumnya sistem operasi memperbolehkan proses atau thread untuk dibatalkan secara asynchronous. Tetapi Pthread API menyediakan deferred cancellation. Hal ini berarti sistem operasi yang mengimplementasikan Pthread API akan mengizinkan deferred cancellation.

## Thread Pools

Pada web server yang menerapkan multithreading ada dua masalah yang timbul: Ukuran waktu yang diperlukan untuk menciptakan thread yang melayani permintaan yang diajukan pada kenyataannya thread dibuang seketika sesudah ia menyelesaikan tugasnya. Pembuatan thread yang tidak terbatas jumlahnya dapat menurunkan performa dari sistem. Solusinya adalah dengan penggunaan Thread Pools, yaitu sekumpulan thread yang mengantri untuk mengerjakan tugas. Cara kerjanya adalah dengan membuat beberapa thread pada proses startup dan menempatkan mereka ke pools, dimana mereka duduk diam dan menunggu untuk bekerja. Jadi, ketika server menerima permintaan, ia akan membangunkan thread dari pool dan jika thread tersedia maka permintaan tersebut akan dilayani. Ketika thread sudah selesai mengerjakan tugasnya maka ia kembali ke pool dan menunggu pekerjaan lainnya. Bila tidak



ada thread yang tersedia pada saat dibutuhkan maka server menunggu sampai ada satu thread yang bebas

### Penjadwalan Thread

Begitu dibuat, thread baru dapat dijalankan dengan berbagai macam penjadwalan. Kebijakan penjadwalanlah yang menentukan setiap proses, di mana proses tersebut akan ditaruh dalam daftar proses sesuai prioritasnya dan bagaimana ia bergerak dalam daftar proses tersebut. Untuk menjadwalkan thread, sistem dengan model multithreading many to many atau many to one menggunakan:

Process Contention Scope (PCS). Pustaka thread menjadwalkan thread pengguna untuk berjalan pada LWP (lightweight process) yang tersedia.

System Contention Scope (SCS). SCS berfungsi untuk memilih satu dari banyak thread, kemudian menjadwalkannya ke satu thread tertentu (CPU/Kernel).

## **BAB III**

### **PENUTUP**

#### **3.1 Kesimpulan**

Proses adalah unit kerja terkecil yang secara individu memiliki sumber daya dan diadalkan oleh sistem operasi. Peran sistem operasi dalam kegiatan proses adalah mengelola semua proses di sistem dan mengalokasikan sumber daya ke proses tersebut, banyak proses yang dijalankan bersamaan, dimana setiap proses mendapat bagian memori dan kendali sendiri-sendiri, sehingga setiap proses memiliki prinsip :

- Independent
- One program at any instant.

Dalam multiprogramming, teknik penanganan proses adalah dengan mengeksekusi satu proses dan secara cepat beralih ke proses lainnya. Thread merupakan sebuah status eksekusi (running, ready, waiting, dll.), sering disebut dengan lightweight process. Merupakan unit dasar dari penggunaan CPU, yang terdiri dari thread\_id, program counter, register set, dan stack. Sebuah thread berbagi code section, data section, dan sumber daya sistem operasi dengan Thread lain yang dimiliki oleh proses yang sama. Sistem operasi telah mendukung proses multithreading. Setiap sistem operasi memiliki konsep tersendiri dalam pengimplementasiannya. Sistem operasi dapat mendukung thread pada tingkatan kernel maupun tingkatan pengguna.

#### **3.2 Saran**

Apabila ada kesalahan dalam penyusunan Makalah ini, saya sebagai penulis sangat mengharapkan kritik dan saran dari pembaca untuk perbaikan tulisan selanjutnya.

