

ABRAHAM SILBERSCHATZ - PETER BAER GALVIN - GREG GAGNE

Operating System:

Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, *Operating System Concepts Essentials*, 2012, 2nd Edition, John Wiley & Sons, Inc.

CHAPTER 3 PART 2: PROCESS MANAGEMENT

PRIJO HANDOKO, S.KOM., M.TI.



An Overview

A **process** can be thought of as a program in execution and will need certain resources — such as CPU time, memory, files, and I/O devices — to accomplish its task. These resources are allocated to the process either when it is created or while it is executing.

Systems consist of a collection of processes: **operating-system processes execute system** and **user processes execute user**.

Although traditionally a process contained only a single *thread* of control as it ran, most modern operating systems now support processes that have multiple threads.

Chapter 3 Part Two: Process Management | 2

Ch. 3: Process Management

Chapter Objectives.

- To introduce the notion of a process — a program in execution, which forms the basis of all computation.
- To describe the various features of processes, including scheduling, creation, and termination.
- To explore interprocess communication using shared memory and message passing.
- To describe communication in client–server systems.

Chapter 3 Part Two: Process Management | 3

Ch. 3: Process Management

Agendas.

- Process Concept
- Process Scheduling
- Operations on Operation
- Interprocess Communication
- Communication in Client – Server Systems

Chapter 3 Part Two: Process Management | 4

Process Concept

The Process.

- Proses adalah kondisi dimana terdapat suatu program yang tengah dieksekusi oleh CPU.
- **Kapankah sebuah program kemudian berubah menjadi sebuah proses?**
- **Active entity & passive entity**
- Selain kode program, sebuah proses juga berisikan:
 - **program counter;**

Process Concept

- **process stack;**
- **data section;** dan
- **heap.**
- **Ada berapakah teknik yang dapat digunakan untuk memuatkan executable file ke dalam memory? Jelaskan!**

Process Concept

Process State.

- Proses dikerjakan melalui serangkaian kondisi (**state**) dan sebuah kejadian dapat merubah **state** proses.
- Kondisi (state) proses adalah:
 - **New**
 - **Running**
 - **Waiting**
 - **Ready**
 - **Terminated.**

Process Concept

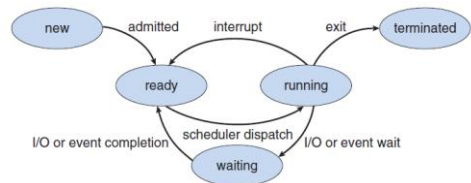
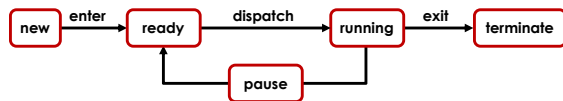


Diagram of Process State.

Process Concept

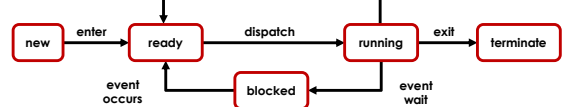
2 States Process Model.



- Kondisi *pause* muncul ketika sebuah proses yang tengah dikerjakan (*running*) mengalami kehabisan waktu pengolahan/eksekusi.

Process Concept

3 States Process Model.

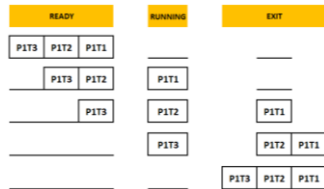


- Kondisi *blocked* muncul ketika sebuah proses yang tengah dikerjakan (*running*) harus menunggu sebuah kejadian (misal proses I/O) selesai dilakukan.

Process Concept

Contoh Kasus.

Jika diketahui sebuah proses, yaitu $A + B * C - D$ akan dikerjakan oleh CPU, maka gambarkan kondisi antrian state ready, state running, dan exit dari awal hingga semua thread selesai diproses!



Process Concept

Latihan 1.

Diketahui *job1* $A * (B - C)$ tengah diproses, ketika *job1* tersebut tengah mengerjakan *thread*-nya yang pertama, *job2* yaitu $X + Y * Z$ melakukan *interrupt* untuk meminta layanan pengolahan.

Berdasarkan kondisi di atas, maka gambarkan kondisi antrian state ready, state running, dan exit dari awal hingga semua thread selesai diproses!

Process Concept

Latihan 2.

Diketahui $job1 A * (B - C)$ tengah diproses, ketika $job1$ tersebut tengah mengerjakan $thread$ -nya yang pertama, $job2$ yaitu $X + Y$ melakukan *interrupt* untuk meminta layanan pengolahan.

Berdasarkan kondisi di atas, maka gambarkan kondisi antrian *state ready*, *state running*, dan *exit* dari awal hingga semua *thread* selesai diproses!

Kondisi:

1. J1T2 dan J2T1 mengalami kondisi *pause*.

Process Concept

Latihan 3.

Diketahui 2 buah *job* berikut:

1. Job 1: $A + B - C$
2. Job 2: $X / Y * Z$

Kondisi:

- Setiap *thread* membutuhkan waktu 2 detik untuk *running*
- T2J2 dan T1J1 mengalami kondisi *blocked* selama 1 detik

Berdasarkan kondisi di atas, maka gambarkan kondisi antrian *state ready*, *state running*, *state blocked*, dan *exit* dari awal hingga semua *thread* selesai diproses!

Process Concept

Process Control Block.

- Setiap proses yang dikerjakan oleh sistem operasi disrepresentasikan sebagai **process control blok** (PCB)
- PCB berisikan: *process state*, *program counter*, *CPU registers*, *CPU-scheduling information*, *memory-management information*, *accounting information*, dan *I/O status information*.

process state
process number
program counter
registers
memory limits
list of open files
...

Process Concept

Threads.

- Proses adalah eksekusi sebuah program.
- Setiap program yang dieksekusi berarti menjalankan sebuah *thread* tertentu.
- **Thread** dikendalikan agar sebuah proses hanya menjalankan sebuah tugas (*task*) tertentu.
- Konsep sistem operasi modern mengizinkan sebuah proses menjalankan banyak *thread* dalam satu waktu (*multithread*).
- Kondisi ini hanya dapat berjalan pada sistem *multicore*.

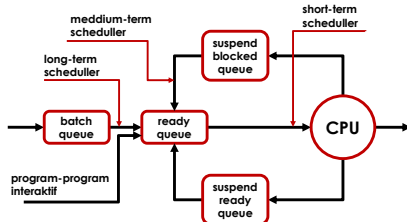
Process Scheduling

- Kumpulan prosedur dan mekanisme yang terdapat dalam SO yang melakukan mengatur dan menyusun urutan pemrosesan proses (*job*) oleh sistem komputer.
- Penjadwalan proses diatur berdasarkan:
 - urutan kedatangan/level prioritas/jumlah waktu pemrosesan/ukuran proses; dan
 - waktu pemroses (quantum/kwanta).
- Penjadwalan proses harus memenuhi beberapa sasaran berikut: *fairness, efficiency, response time, turn around time, dan throughput.*

Process Scheduling

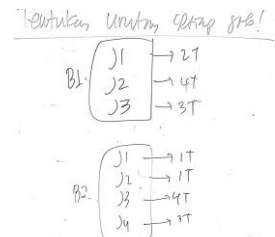
- Penjadwalan meliputi:
 1. **Jangka Pendek** (*short-term scheduler*), menjadwalkan alokasi memori untuk *job-job ready*
 2. **Jangka Menengah** (*medium-term scheduler*), mengendalikan transisi *suspended* → *ready job-job yang di-swapping*
 3. **Jangka Panjang** (*long-term scheduler*), mengelola antrian batch dan memilih batch berikutnya yang akan diproses/dieksekusi.

Process Scheduling



Process Scheduling

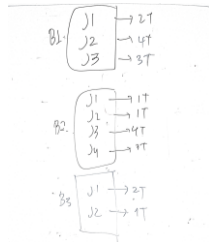
Latihan 4.
Tentukan urutan *job-job* dari *batch-batch* proses berikut dalam *state ready!*



Process Scheduling

Latihan 5.

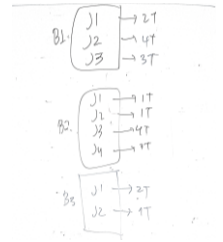
Tentukan urutan *job-job* dari *batch-batch* proses berikut dalam *state ready*!



Process Scheduling

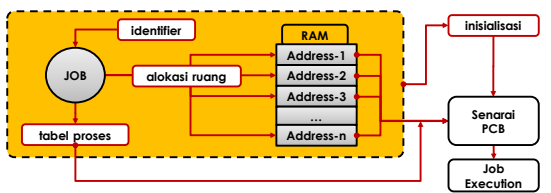
Latihan 6.

Perhatikan *batch* proses berikut. Jika J2T2 dan J3T3 mengalami *blocked* sedangkan J3T1 dan J2T1 mengalami *timeout* maka tentukan urutan proses *job-job* tersebut!



Operation on Process

Process Creation.



Operation on Process

Process Diversion.

• Beberapa penyebab terjadinya pengalihan proses:

1. System interrupt

- Clock interrupt
- I/O interrupt
- Page/memory fault

2. Trap

Interrupsi *by error*

3. Supervisor Call

Pengaktifan kembali bagian OS

Operation on Process

Process Termination.

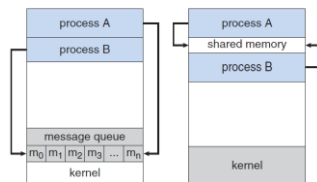
- Sebuah proses dikatakan berakhir (*terminated*), apabila proses telah mengeksekusi pernyataan (*statement*) terakhir yang terdapat dalam program dan *system call* dihapuskan.

Interprocess Communication

- Proses eksekusi dapat berupa *independent processes* atau *cooperating processes*.
- Beberapa alasan *cooperating process* memungkinkan dilakukan:
 - *information sharing*, menyediakan lingkungan sistem yang memungkinkan akses ke suatu informasi;
 - *computing speedup*, membagi *task* kedalam *subtask* dan setiap *subtask* akan dieksekusi secara paralel;
 - *modularity*, membagi fungsi sistem ke dalam proses atau *thread* yang terpisah.
 - *convenience*.

Interprocess Communication

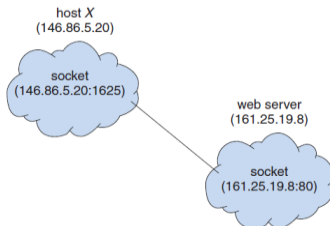
- *Cooperating processes* membutuhkan *interprocess communication (IPC)*.
- Terdapat 2 model dasar dalam *interprocess communication*:
 - *shared memory model*;
 - *message-passing model*



Communication in Client-Server Systems

- Terdapat 3 strategi dalam komunikasi pada sistem **client-server**
 1. **Sockets**
 2. **Remote Procedure Call (RPC)**
 3. **Pipes**
- **Sockets.**
 - Socket didefinisikan sebagai *end-point* untuk komunikasi dan merupakan gabungan antara *IP address* dan nomor *port*.

Communication in Client-Server Systems



Chapter 3 Part Two: Process Management | 29

Communication in Client-Server Systems

Remote Procedure Call (RPC).

- RPC dirancang sebagai suatu mekanisme pemanggilan prosedur (*procedure-call mechanism*) yang digunakan pada sistem yang terkoneksi dalam jaringan.

Pipes.

- *Pipes* bertindak sebagai saluran yang memungkinkan 2 atau lebih proses untuk saling berkomunikasi.

Chapter 3 Part Two: Process Management | 30

ABRAHAM SILBERSCHATZ - PETER BAER GALVIN - GREG GAGNE

Operating System:

Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, *Operating System Concepts Essentials*, 2012, 2nd Edition, John Wiley & Sons, Inc.

CHAPTER 3 PART 2: PROCESS MANAGEMENT

TERIMA KASIH