

SISTEM OPERASI

SISTEM OPERASI

Semua Tentang Sistem Operasi

Rolly Maulana Awangga



A JOHN WILEY & SONS, INC., PUBLICATION

Copyright ©2018 by John Wiley & Sons, Inc. All rights reserved.

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.
Published simultaneously in Canada.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600, or on the web at www.copyright.com. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008.

Limit of Liability/Disclaimer of Warranty: While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. Neither the publisher nor author shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

For general information on our other products and services please contact our Customer Care Department with the U.S. at 877-762-2974, outside the U.S. at 317-572-3993 or fax 317-572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print, however, may not be available in electronic format.

Library of Congress Cataloging-in-Publication Data:

Web Service / Rolly Maulana Awangga
Printed in the United States of America.

10 9 8 7 6 5 4 3 2 1

For my family

CONTENTS IN BRIEF

PART I PENGENALAN SISTEM OPERASI

1 OS Semaphore	3
2 Proses OS	7
3 Sistem Operasi	13
4 Starvation	23
5 DeadLock	27
6 Baudrate	31
7 Serial Communication di Linux	35
8 SERIALCOMWINDOWS	41

CONTENTS

List of Figures	xiii
List of Tables	xv

PART I PENGENALAN SISTEM OPERASI

1	OS Semaphore	3
1.1	System Operasi Semaphore	3
1.1.1	Definisi	3
1.1.2	Prinsip Semaphore	4
1.1.3	Kelemahan Semaphore	5
1.1.4	Semantik dan Impelementasi	5
1.1.5	Prinsip Semaphore	6
1.1.6	Kelemahan Semaphore	6
2	Proses OS	7
2.1	proses	7
2.1.1	Proses	7
2.1.2	Istilah yang berkaitan dengan proses dalam sistem operasi	7
		ix

2.1.3	Fungsi fungsi sistem operasi	8
2.1.4	jenis jenis sistem operasi pada komputer	8
2.1.5	Status proses	9
2.1.6	Contoh dari status proses	10
2.1.7	Proses Control Block/Blocked	11
2.1.8	Trace dan Dispatcher	11
3	Sistem Operasi	13
3.1	Sistem Operasi	13
3.1.1	Definisi	13
3.1.2	Fungsi	14
3.1.3	Jenis sistem operasi	15
3.1.4	Single dan multi-user	15
3.1.5	Pendistribusian	15
3.1.6	Sejarah	15
3.1.7	Mikrokomputer	17
3.1.8	Berkeley Software Distribution	18
3.1.9	Linux	19
3.1.10	macOS	19
3.1.11	Microsoft Windows	20
4	Starvation	23
4.1	Starvation	23
4.1.1	Definisi	23
4.1.2	Algoritma Starvation	24
4.2	Teknik menghadapi starvation	24
4.2.1	Menghindari Starvation	24
4.2.2	Ilustrasi	25
4.3	Menghambat Starvation dengan Disclosed	25
4.4	Ilustrasi	25
5	DeadLock	27
5.1	DEADLOCK	27
5.1.1	Deadlock	27
5.1.2	Masalah Deadlock dan Metode Penanganan Deadlock	28
5.1.3	Deadlock Detection	29
5.1.4	Beberapa hal yang terjadi ketika mendeteksi adanya deadlock	29

5.1.5	Beberapa jalan untuk kembali dari deadlock	29
6	Baudrate	31
6.1	Baud Rate	31
6.1.1	Pengertian Baud Rate	31
6.1.2	Konsep Baud Rate	31
6.1.3	Hubungan baud rate dengan bit rate	33
6.1.4	Fungsi Baud Rate	33
6.1.5	Perbedaan Baud Rate Dengan Bit Rate	33
6.1.6	Framing Data	34
6.1.7	Contoh Pengiriman Data	34
7	Serial Communication di Linux	35
7.1	Komunikasi Serial pada Linux	35
7.1.1	Konsep Dasar Komunikasi Serial	35
7.1.2	Interprocess Communication	36
7.1.3	Fungsi utama komunikasi serial	36
7.1.4	contoh fungsi utama	37
7.1.5	dua metode komunikasi serial	37
7.1.6	macam-macam perintah terminal linux	37
7.1.7	perbedaan port USB dan port serial	37
7.1.8	cara menggunakan komunikasi serial menggunakan port serial menjadi port USB	37
7.1.9	Koneksi Linux ke Serial Port	37
8	SERIALCOMWINDOWS	41
8.1	Serial Com Windows	41
8.1.1	Membuka Port	41
8.1.2	I atau O tumpang tindih	42
8.1.3	Membaca dan menulis	44
8.1.4	Serial Status	46
8.1.5	Communications Events	47
8.1.6	Flow Control	47

LIST OF FIGURES

1.1	Semaphore	3
2.1	Gambar Status Proses	10
3.1	Sistem Operasi	14
4.1	Deadlock vs Starvation.	23
5.1	Gambar Deadlock	28
6.1	Serial Frame	32
7.1	Gambar Status Serial	38
7.2	Gambar minicom	39
8.1	Serial Com Windows	43

LIST OF TABLES

3.1	FAT	21
-----	-----	----

PART I

PENGENALAN SISTEM OPERASI

CHAPTER 1

OS SEMAPHORE

1.1 System Operasi Semaphore

1.1.1 Definisi

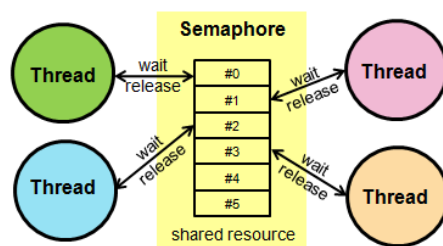


Figure 1.1 Semaphore

Semaphore merupakan salah satu teknik sinyal pada OS yang paling sederhana, dan merupakan konsep yang penting dalam OS desain, di mana nilai integer digunakan untuk memberi sinyal antar proses. Hanya tiga operasi yang mungkin di-

Sistem Operasi, pre-release.

By Rolly Maulana Awangga Copyright © 2019 John Wiley & Sons, Inc.

lakukan pada semaphores, yang semuanya adalah atom: inisialisasi, keturunan, dan peningkatan. Pengurangan operasi dapat menyebabkan proses yang diblokir, dan peningkatan operasi yang sedang berlangsung dapat mengakibatkan pemblokiran suatu proses. Semaphore pada system operasi merupakan sebuah variabel bertipe integer. Di kehidupan nyata, semaphore merupakan sistem sinyal yang digunakan untuk memberi sinyal atau tanda dan berkomunikasi secara visual. Semafor juga merupakan struktur data dalam bahasa komputer yang digunakan untuk menyinkronkan suatu proses, yaitu untuk memecahkan masalah di mana masalahnya lebih dari satu proses atau bisa seperti thread yang akan dijalankan secara bersamaan dan harus diatur urutan kerja. Semaphore dibuat oleh Edsger Dijkstra dan pertama kali digunakan dalam sistem operasi. Nilai semaphore diinisialisasi dengan jumlah sumber daya yang dikontrol oleh pengguna. Dalam kasus khusus di mana ada sumber daya bersama, semaphore disebut semaphore biner. Semaphore adalah solusi klasik dari dining philosophers problem, meskipun itu tidak mencegah deadlock. Pada software semaphore, semaphore merupakan variabel yang bertipe data integer tetapi tidak termasuk pada data yang sedang dilakukan inisialisasi, yang hanya dapat diakses melalui dua operasi standar, yaitu increment dan decrement. Semaphore bisa digunakan untuk menyelesaikan masalah sinkronisasi secara umum, berdasarkan jenisnya. Semaphore hanya memiliki nilai 1 atau 0, atau lebih dari sama dengan 0. Konsep semaphore pertama kali diajukan idenya oleh Edsger Dijkstra pada tahun 1967. Semaphore memiliki dua jenis, yaitu, Biner semaphore dan counting semaphore. Biner semaphore tidak bisa memiliki semua jenis integer tetapi hanya memiliki 2 nilai yaitu 1 atau 0, Sering juga disebut sebagai semaphore primitive. Sedangkan Counting semaphore memiliki nilai 0, 1, sampai seterusnya atau integer lainnya. Banyak sistem operasi yang tidak secara langsung menggunakan semaphore jenis ini, namun lebih banyak yang memanfaatkan semaphore jenis biner semaphore. Pada semaphore ini harus diketahui bahwa, ada beberapa jenis dari counting semaphore yang salah satu jenisnya adalah semafor yang tidak bisa mencapai nilai negatif dan jenis yang lain adalah semaphore yang dapat mencapai nilai negatif. Solusi dari Pembuatan Counting Semaphore adalah Binary Semaphore. Pembuatan counting semaphore banyak dilakukan para programmer untuk memenuhi alat sinkronisasi yang sesuai dengannya.

Operasi standarnya dalam bahasa pemrograman C :

```
void kunci(int sem_value) {
while(sem_value <= 0);
sem_value;
}
void buka(int sem_value) {
sem_value++;
}
```

1.1.2 Prinsip Semaphore

1. Suatu proses yang berbeda bisa berkaitan dengan memanfaatkan sinyal - sinyal.

2. Suatu proses dapat dihentikan oleh proses yang lain.
3. Semaphore bertipe data integer yang diakses oleh dua operasi atomik standar (wait dan signal).
4. Ada dua operasi di semaphore (Down dan UP). Yang nama aslinya : P dan V.

1.1.3 Kelemahan Semaphore

1. Semaphore termasuk Low Level.
2. Dikarenakan semaphore tersebar di dalam seluruh program maka kita akan kesulitan dalam pemeliharannya.
3. Jika kita menghapus wait akan mengakibatkan nonmutual exclusion.
4. Jika kita menghapus signal akan mengakibatkan deadlock.
5. Jika terjadi deadlock akan sulit untuk dideteksi.

1.1.4 Semantik dan Implementasi

Menghitung semaphores dilengkapi dengan dua operasi, secara historis dilambangkan sebagai P dan V (lihat Nama operasi untuk nama alternatif). Operasi V menambahkan semaphore S, dan operasi P menurunkannya.

Nilai semaphore S adalah jumlah unit sumber daya yang saat ini tersedia. Operasi P membuang waktu atau tidur sampai sumber daya yang dilindungi oleh semaphore menjadi tersedia, pada saat itu sumber daya segera diklaim. Operasi V adalah kebalikannya: ia membuat sumber daya tersedia lagi setelah proses selesai menggunakannya. Satu properti penting dari semaphore S adalah bahwa nilainya tidak dapat diubah kecuali dengan menggunakan operasi V dan P.

Cara sederhana untuk memahami operasi tunggu (P) dan sinyal (V) adalah:

- menunggu: Jika nilai variabel semaphore tidak negatif, turunkan dengan 1. Jika variabel semaphore sekarang negatif, proses menunggu eksekusi diblokir (yaitu, ditambahkan ke antrian semaphore) sampai nilainya lebih besar atau sama dengan 1. Jika tidak, proses terus berjalan, setelah menggunakan satu unit sumber daya.
- sinyal: Menambah nilai semaphore variabel dengan 1. Setelah kenaikan, jika nilai pre-increment negatif (berarti ada proses menunggu sumber daya), ia mentransfer proses yang diblokir dari antrian menunggu semaphore ke antrian siap.

Banyak sistem operasi menyediakan primitif semaphore yang efisien yang membuka blokir proses menunggu ketika semaphore bertambah. Ini berarti bahwa proses tidak membuang waktu untuk memeriksa nilai semaphore yang tidak perlu. Konsep penghitungan semaphore dapat diperpanjang dengan kemampuan untuk mengklaim

atau mengembalikan lebih dari satu unit dari semaphore, teknik yang diterapkan di Unix. Operasi V dan P yang dimodifikasi adalah sebagai berikut, menggunakan tanda kurung siku untuk menunjukkan operasi atom, yaitu operasi yang tampak terpisah dari perspektif proses lain:

Semaphore pada system operasi merupakan sebuah variabel bertipe integer. Di kehidupan nyata, semaphore merupakan sistem sinyal yang digunakan untuk memberi sinyal atau tanda dan berkomunikasi secara visual. Pada software semaphore, semaphore merupakan variabel yang bertipe data integer tetapi tidak termasuk pada data yang sedang dilakukan inisialisasi, yang hanya dapat diakses melalui dua operasi standar, yaitu increment dan decrement. Semaphore bisa digunakan untuk menyelesaikan masalah sinkronisasi secara umum, berdasarkan jenisnya. Semaphore hanya memiliki nilai 1 atau 0, atau lebih dari sama dengan 0. Konsep semaphore pertama kali diajukan idenya oleh Edsger Dijkstra pada tahun 1967. Semaphore memiliki dua jenis, yaitu, Biner semaphore dan counting semaphore. Biner semaphore hanya memiliki nilai 1 atau 0, Sering juga disebut sebagai semaphore primitive. Sedangkan Counting semaphore memiliki nilai 0, 1, sampai seterusnya atau integer lainnya. Banyak sistem operasi yang tidak secara langsung menggunakan semaphore jenis ini, namun lebih banyak yang memanfaatkan semaphore jenis biner semaphore

1.1.5 Prinsip Semaphore

1. Suatu proses yang berbeda bisa berkaitan dengan memanfaatkan sinyal - sinyal.
2. Suatu proses dapat dihentikan oleh proses lainnya.
3. Semaphore bertipe data integer yang diakses oleh dua operasi atomik standar (wait dan signal).
4. Ada dua operasi di semaphore (Down dan UP). Yang nama aslinya : P dan V.

1.1.6 Kelemahan Semaphore

1. Semaphore termasuk Low Level.
2. Dikarenakan semaphore tersebar di dalam seluruh program maka kita akan kesulitan dalam pemeliharaannya.
3. Jika kita menghapus wait akan mengakibatkan nonmutual exclusion.
4. Jika kita menghapus signal akan mengakibatkan deadlock.
5. Jika terjadi deadlock akan sulit untuk dideteksi.

[?] [?] [?]

CHAPTER 2

OS PROSES

2.1 proses

2.1.1 Proses

Proses adalah sebuah program yang sedang dieksekusi. Sedangkan program adalah merupakan kumpulan-kumpulan dari suatu instruksi yang sudah ditulis ke dalam bahasa yang dapat dimengerti oleh sistem operasi. Proses berisi tentang sebuah instruksi dan sebuah data. program counter dan seluruh register pemroses, stack ini berisi data sementara contohnya seperti alamat pengiriman, parameter rutin dan variabel lokal. Sistem operasi diharuskan untuk mengelola semua proses di dalam sistem tersebut dan mengalokasikan sumber daya ke sebuah proses-proses sesuai dengan kebijaksanaan untuk memenuhi sasaran sistem

2.1.2 Istilah yang berkaitan dengan proses dalam sistem operasi

- Multiprogramming Multiprogramming adalah istilah teknologi informasi dengan menggunakan bahasa Inggris yang baik mengacu kepada sebuah metode dimana banyak sebuah pekerjaan atau yang dikenal juga sebagai proses dan

gan diolah dengan menggunakan sumber daya CPU yang sama. Dalam sistem ini sebuah program dijalankan dalam CPU sampai terjadi interupsi. pada saat proses interupsi terjadi maka program yang dimuat dalam memori akan di jalannkan sampai proses interupsi.ketika proses interupsi selesai maka kontrol dikembalikan pada program yang sebelumnya. Contohnya proses ketika membuka program aplikasi yang berbeda dalam satu waktu.

- Time-Sharing adalah suatu metode yang digunakan dalam sistem operasi yang memungkinkan sejumlah pemakai dapat berinteraksi dengan proses, setiap proses akan ditangani oleh CPU secara bergantian dalam jumlah waktu yang sama.
- Multitasking adalah suatu proses yang memungkinkan seorang pengguna dapat menjalankan sejumlah program secara bersamaan. contohnya membuka word dan excel secara bersamaan.
- Multiprocessing kemampuan komputer untuk melakukan beberapa proses dengan waktu yang bersamaan, dibantu dengan keberadaan teknologi yang berbasis multiprocessor. Contohnya seperti computer server.
- Distributed processing/computing Mengerjakan semua proses pengolahan data secara bersamaan antara komputer pusat dengan beberapa komputer yang lebih kecil dan saling berhubungan denan melalui jalur komunikasi. Contohnya komputer yang dirancang untuk melaksanakan tugas-tugas proyek.

2.1.3 Fungsi fungsi sistem operasi

1. Resource manager, pengolahan sumber daya dan mengalokasikan.
2. Interface atau tatap muka, sebagai perantara antara pengguna dengan perangkat keras.
3. Coordinator, mengkoordinasi fasilitas sehingga aktifitas yang kompleks dapat diatur..

2.1.4 jenis jenis sistem operasi pada komputer

1. Windows, merupakan pengembangan dari sistem operasi DOS. windows juga mudah untuk dipelajari.
2. Mac OS, merupakan sistem operasi yang diciptakan oleh Apple. Mac OS memiliki tingkat keamanan yang tinggi.
3. Linux, memiliki kestabilan yang baik, yang sering digunakan sebagai sistem operasi pada server.
4. Android, merupakan sistem operasi pada smartphone. android sama seperti Linux, yaitu mudah dikembangkan.

2.1.5 Status proses

Terdapat 5 macam jenis status yang mungkin dimiliki oleh suatu proses :

1. New, yaitu status yang dimiliki pada saat proses baru saja terjadi.
2. Ready, yaitu status dimana proses siap untuk dieksekusi pada giliran berikutnya.
3. Running, yaitu status suatu proses dimana saat ini proses tersebut sedang dieksekusi oleh prosesor
4. Waiting, yaitu status dimana proses yang tidak bisa dijalankan di saat prosesor sudah siap, status yang dimiliki pada saat proses menunggu suatu sebuah event seperti I/O
5. Terminated, yaitu status yang dimiliki pada saat proses telah selesai dieksekusi

Berikut ini adalah proses dari ke-5 status proses di atas :

1. New ke Ready Pertama Status dibuat lalu setelah itu , status akan memasuki proses ready dan siap untuk memasuki proses selanjutnya.
2. Ready ke running Di saat sedang memilih proses yang akan dioperasikan, sistem operasi akan memilih salah satu proses yang berada didalam keadaan status ready.
3. Running ke waiting Suatu proses dimasukkan dalam keadaan status waiting jika proses itu meminta sesuatu yang dapat menyebabkannya harus menunggu. Sebuah request ke sistem operasi yang pada umumnya merupakan bentuk panggilan dari layanan sistem (panggilan dari program yang sedang beroperasi ke dalam suatu prosedur yang merupakan bagian kode sistem operasi) misalnya seperti sebuah proses yang bisa meminta suatu layanan dari sistem operasi yang tidak siap dilakukan oleh sistem operasi dengan segera. Atau proses yang bisa menginisiasi suatu aksi, seperti misalnya operasi I/O, yang seharusnya bisa diselesaikan sebelum proses itu melanjutkan operasinya. Pada saat proses saling melakukan komunikasi dengan proses yang lainnya, suatu proses bisa diblokir jika sedang menunggu proses lainnya untuk menyediakan input.
4. Running ke ready Pada umumnya alasan transisi ini ialah dimana sebuah proses yang lagi berjalan sudah mencapai batas waktu maksimum yang telah diizinkan bagi instruksi yang tidak diinterupsi. Terdapat beberapa alasan yang menyebabkan transisi ini terjadi, yang tidak diimplementasikan oleh setiap sistem operasi. Misalnya apabila sistem operasi meng-assign tingkat prioritas yang berbeda pada beberapa proses yang berlainan, suatu proses bisa diambil lebih dulu.
5. Waiting ke ready Apabila suatu proses dalam keadaan status waiting sudah selesai dalam mendapatkan sumber daya, seperti file atau bagian virtual memori bagi pakai atau juga sudah selesai setelah menunggu proses yang lainnya untuk menyediakan input atau sudah selesai dalam menunggu pesan lainnya.

6. Runing ke finish(terminated) proses yang sedang berjalan dihentikan oleh Sistem Operasi jika proses itu telah selesai atau tidak jadi dieksekusi. Hal ini terjadi dikarenakan jika proses induknya sendiri telah berhenti.

Dan berikut ini adalah diagram dari ke-5 status proses tadi :

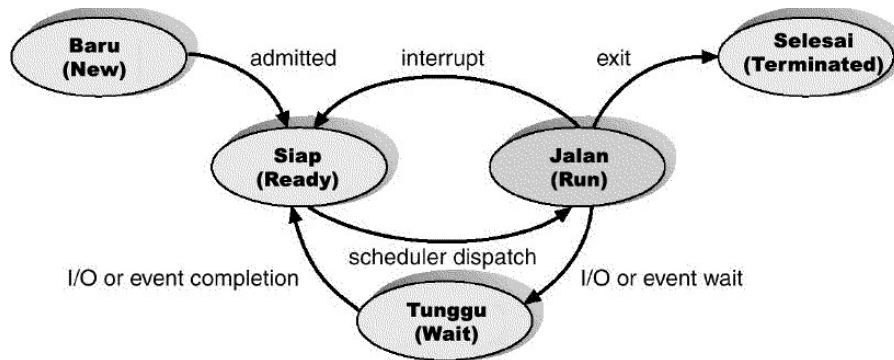


Figure 2.1 Gambar Status Proses

2.1

2.1.6 Contoh dari status proses

Setiap proses pasti mempunyai status yang harus diperhatikan oleh sistem operasi yang akan dicatat dalam berbagai macam tabel yang saling berhubungan, yaitu :

- Tabel informasi manajemen memori, Digunakan untuk menjaga keutuhan dari suatu memori utama dan memori sekunder yang akan menyimpan suatu informasi.
- Tabel informasi manajemen masukan atau keluaran, Digunakan untuk melakukan pengelolaan sebuah perangkat masukan atau keluaran, yang dimana perangkat tersebut akan digunakan oleh proses yang tertentu, sehingga harus dijaga supaya proses yang lainnya tidak akan memakainya. Sistem operasi harus mengetahui status operasi masukan atau keluaran dan lokasi suatu memori utama yang akan digunakan untuk melakukan transfer data.
- Tabel informasi sistem file, Tabel dimana yang berisikan mengenai informasi lokasi pada memori sekunder, informasi ekstensi file, informasi status pada saat itu dan menyimpan atribut-atribut file yang lainnya.
- Tabel proses, Digunakan untuk mengelola informasi suatu proses pada sistem operasi, yang berlokasi di memori, atribut proses dan status lainnya.

2.1.7 Proses Control Block/Blocked

Sebuah struktur data yang dipakai oleh Sistem Operasi untuk mengelola sebuah proses. Hampir semua Sistem Operasi yang modern telah menggunakan PCB (Process Control Block) namun strukturnya yang berbeda-beda pada setiap Sistem Operasi tersebut. PCB (Process Control Block) juga memiliki informasi yang berhubungan dengan proses, ialah: tanda pengenal bagi sebuah proses (Process ID) yang sangat unik dan akan menjadi status proses, nomor identitas, prioritas eksekusi proses dan semua informasi lokasi proses dalam memori. Prioritas yang dimiliki oleh sebuah proses merupakan sebuah nilai atau besaran yang akan menunjukkan seberapa sering proses harus dieksekusi oleh prosesor. Sebuah proses yang mempunyai nilai prioritas yang cenderung lebih tinggi, akan lebih sering dieksekusi atau dieksekusi terlebih dulu jika dibandingkan dengan proses yang mempunyai prioritas yang lebih rendah. Sebuah PCB (Process Control Block) ditunjukkan dalam tabel berikut :

PCB dibagi menjadi 3 kelompok yaitu :

- Process identification data pasti akan selalu mengikutsertakan suatu identifier yang sangat unik untuk prosesnya (hampir selalu mempunyai nilai integer) dan pada sebuah sistem multiuser-multitasking, data yang contohnya seperti identifier grup pengguna, identifier proses, identifier pengguna, dan yang lainnya. Proses ini sangat relevan, karena itu sering dipakai untuk referensi silang tabel sistem operasi, misalnya seperti memungkinkan untuk mengidentifikasi sebuah proses yang menggunakan device I/O, atau daerah memori.
- Processor state data potongan-potongan dari informasi yang mengartidefinisikan status dari sebuah proses ketika proses tersebut ditangguhkan, dan memungkinkan sistem operasi untuk melakukan restart proses pada akhirnya dan akan masih dapat mengeksekusinya dengan benar. Hal ini akan selalu mengikutsertakan isi dari register CPU tujuan.
- Process control data digunakan oleh sistem operasi untuk mengelola proses itu sendiri.

2.1.8 Trace dan Dispatcher

Trace adalah daftar urutan instruksi yang sudah dijalankan pada suatu proses. Sedangkan dispatcher adalah program kecil yang telah mengatur proses mana yang akan dijalankan oleh sebuah prosesor. Dirangkum dari makalah [?] Dirangkum dari makalah [?]

Pointer	State Proses
	Nomor Proses
	Program Counter
	Registers
	Batas Memori
	Daftar berkas yang telah dibuka

CHAPTER 3

OS SISTEM OPERASI

3.1 Sistem Operasi

3.1.1 Definisi

1. Sistem operasi (OS) adalah sistem perangkat lunak yang mengelola hardware dan sumber daya perangkat lunak dengan menyediakan pelayanan secara umum untuk program komputer.
2. Sistem operasi berbagi waktu menjadwalkan tugas untuk penggunaan sistem yang efisien dan mungkin juga termasuk perangkat lunak akuntansi untuk alokasi biaya waktu prosesor, penyimpanan massal, pencetakan, dan sumber daya lainnya.
3. Untuk fungsi perangkat keras seperti input dan output dan alokasi memori, sistem operasi bertindak sebagai perantara antara program dan perangkat keras komputer, meskipun kode aplikasi biasanya dijalankan langsung oleh perangkat keras dan sering membuat panggilan sistem ke fungsi OS atau terganggu oleh saya . Sistem operasi banyak ditemukan pada perangkat komputer, telepon seluler dan permainan konsol video ke server web dan superkomputer.

4. Sistem operasi berbagi waktu menjadwalkan tugas untuk penggunaan sistem yang efisien dan mungkin juga termasuk perangkat lunak akuntansi untuk alokasi biaya waktu prosesor, penyimpanan massal, pencetakan, dan sumber daya lainnya.
5. Untuk fungsi perangkat keras seperti input dan output dan alokasi memori, sistem operasi bertindak sebagai perantara antara program dan perangkat keras komputer, meskipun kode aplikasi biasanya dijalankan langsung oleh perangkat keras dan sering membuat panggilan sistem ke fungsi OS atau terganggu olehnya. Sistem operasi banyak ditemukan pada perangkat komputer, telepon seluler dan konsol permainan video ke server web dan superkomputer.
6. Sistem operasi desktop yang dominan adalah Microsoft Windows dengan pangsa pasar sekitar 82,74 persen, macOS oleh Apple Inc. berada di tempat kedua 13,23 persen, dan varietas Linux secara kolektif berada di tempat ketiga 1,57 persen. Di sektor seluler (gabungan ponsel dan tablet), penggunaan pada tahun 2017 adalah hingga 70 persen dari Google Android dan menurut data kuartal ketiga 2016, Android pada smartphone dominan dengan 87,5 persen dan tingkat pertumbuhan 10,3 persen per tahun, diikuti oleh Apple iOS dengan 12,1 persen dan penurunan per tahun di pangsa pasar 5,2 persen, sementara jumlah sistem operasi lainnya hanya 0,3 persen. Distribusi Linux dominan di sektor server dan superkomputer. Kelas khusus lainnya dari sistem operasi, seperti embedded dan sistem real-time, ada untuk banyak aplikasi.



Figure 3.1 Sistem Operasi

3.1.2 Fungsi

1. fungsi dari sistem operasi ini sendiri yaitu sebagai penghubung antara hardware dan software.
2. sebagai wadah/tempat pengelolaan daya dan pengalokasian data.
3. sebagai penerjemah bahasa biner lalu ditampilkan ke monitor berupa kombinasi teks dan grafis.

3.1.3 Jenis sistem operasi

3.1.3.1 *Single dan multi-tasking*

1. Sistem satu tugas hanya dapat menjalankan satu program dalam satu waktu, sementara sistem operasi multitasking memungkinkan lebih dari satu program berjalan dalam konkurensi. Ini dicapai dengan time-sharing, di mana waktu prosesor yang tersedia dibagi antara beberapa proses. Proses ini masing-masing terganggu berulang kali dalam irisan waktu oleh subsistem penjadwalan tugas dari sistem operasi. Multi-tasking dapat dicirikan dalam tipe preemptif dan kooperatif. Dalam preemptive multitasking, sistem operasi memotong waktu CPU dan mendedikasikan slot untuk masing-masing program. Sistem operasi mirip Unix, seperti Solaris dan Linux serta non-Unix-like, seperti AmigaOS mendukung preemptive multitasking. Multitasking kooperatif dicapai dengan mengandalkan pada setiap proses untuk menyediakan waktu untuk proses lain dengan cara yang ditentukan. Versi 16-bit Microsoft Windows menggunakan multi-tasking kooperatif. Versi 32-bit dari Windows NT dan Win9x, menggunakan preemptive multi-tasking.

3.1.4 Single dan multi-user

1. Sistem operasi pengguna tunggal tidak memiliki fasilitas untuk membedakan pengguna, tetapi dapat memungkinkan beberapa program berjalan bersama-sama. Sistem operasi multi-pengguna memperluas konsep dasar multi-tasking dengan fasilitas yang mengidentifikasi proses dan sumber daya, seperti ruang disk, milik beberapa pengguna, dan sistem memungkinkan banyak pengguna untuk berinteraksi dengan sistem pada saat yang bersamaan.
2. Sistem operasi berbagi waktu menjadwalkan tugas untuk penggunaan sistem yang efisien dan mungkin juga termasuk perangkat lunak akuntansi untuk alokasi biaya waktu prosesor, penyimpanan massal, pencetakan, dan sumber daya lainnya untuk banyak pengguna.

3.1.5 Pendistribusian

1. Sistem operasi terdistribusi mengelola sekelompok komputer yang berbeda dan membuatnya tampak sebagai komputer tunggal. Pengembangan jaringan komputer yang dapat dihubungkan dan berkomunikasi satu sama lain menimbulkan komputasi terdistribusi. Komputasi terdistribusi dilakukan pada lebih dari satu mesin. Ketika komputer dalam kelompok bekerja dalam kerja sama, mereka membentuk sistem terdistribusi.

3.1.6 Sejarah

1. Komputer awal dibangun untuk melakukan serangkaian tugas tunggal, seperti kalkulator. Fitur-fitur sistem operasi dasar dikembangkan pada tahun 1950-

an, seperti fungsi monitor penduduk yang dapat secara otomatis menjalankan program yang berbeda secara berurutan untuk mempercepat pemrosesan. Sistem operasi tidak ada dalam bentuk modern dan lebih kompleks hingga awal 1960-an. Fitur perangkat keras ditambahkan, yang memungkinkan penggunaan pustaka runtime, interupsi, dan pemrosesan paralel. Ketika komputer pribadi menjadi populer pada tahun 1980-an, sistem operasi dibuat untuk mereka yang serupa dalam konsep untuk yang digunakan pada komputer yang lebih besar.

2. Pada 1940-an, sistem digital elektronik paling awal tidak memiliki sistem operasi. Sistem elektronik saat ini diprogram pada deretan switch mekanis atau oleh kabel jumper pada papan steker. Ini adalah sistem tujuan khusus yang, misalnya, menghasilkan tabel balistik untuk militer atau mengontrol pencetakan cek gaji dari data pada kartu kertas berlubang. Setelah komputer tujuan umum yang dapat diprogram diciptakan, bahasa mesin (terdiri dari string digit biner 0 dan 1 pada pita kertas berlubang) diperkenalkan yang mempercepat proses pemrograman.
3. Pada awal 1950-an, komputer hanya dapat menjalankan satu program dalam satu waktu. Setiap pengguna memiliki satu-satunya penggunaan komputer untuk jangka waktu terbatas dan akan tiba pada waktu yang dijadwalkan dengan program dan data pada kartu kertas berlubang atau pita berlubang. Program akan dimuat ke dalam mesin, dan mesin akan diatur untuk bekerja sampai program selesai atau crash. Program umumnya dapat di-debug melalui panel depan menggunakan saklar beralih dan lampu panel. Dikatakan bahwa Alan Turing adalah seorang ahli dalam mesin Manchester Mark 1 awal ini, dan dia sudah mendapatkan konsep primitif dari sistem operasi dari prinsip-prinsip mesin Turing universal.
4. Kemudian mesin datang dengan perpustakaan program, yang akan dikaitkan dengan program pengguna untuk membantu dalam operasi seperti input dan output dan menghasilkan kode komputer dari kode simbolik yang dapat dibaca manusia. Ini adalah awal dari sistem operasi modern. Namun, mesin masih menjalankan pekerjaan tunggal pada suatu waktu. Di Cambridge University di Inggris, antrian pekerjaan pada suatu waktu adalah garis pencucian (garis pakaian) dari mana kaset digantung dengan pakaian warna berbeda untuk menunjukkan prioritas pekerjaan.
5. Pada 1940-an, sistem digital elektronik paling awal tidak memiliki sistem operasi. Sistem elektronik saat ini diprogram pada deretan switch mekanis atau oleh kabel jumper pada papan steker. Ini adalah sistem tujuan khusus yang, misalnya, menghasilkan tabel balistik untuk militer atau mengontrol pencetakan cek gaji dari data pada kartu kertas berlubang. Setelah komputer tujuan umum yang dapat diprogram diciptakan, bahasa mesin (terdiri dari string digit biner 0 dan 1 pada pita kertas berlubang) diperkenalkan yang mempercepat proses pemrograman.

6. Pada awal 1950-an, komputer hanya dapat menjalankan satu program dalam satu waktu. Setiap pengguna memiliki satu-satunya penggunaan komputer untuk jangka waktu terbatas dan akan tiba pada waktu yang dijadwalkan dengan program dan data pada kartu kertas berlubang atau pita berlubang. Program akan dimuat ke dalam mesin, dan mesin akan diatur untuk bekerja sampai program selesai atau crash. Program umumnya dapat di-debug melalui panel depan menggunakan saklar beralih dan lampu panel. Dikatakan bahwa Alan Turing adalah seorang ahli dalam mesin Manchester Mark 1 awal ini, dan dia sudah mendapatkan konsep primitif dari sistem operasi dari prinsip-prinsip mesin Turing universal.
7. Kemudian mesin datang dengan perpustakaan program, yang akan dikaitkan dengan program pengguna untuk membantu dalam operasi seperti input dan output dan menghasilkan kode komputer dari kode simbolik yang dapat dibaca manusia. Ini adalah awal dari sistem operasi modern. Namun, mesin masih menjalankan pekerjaan tunggal pada suatu waktu. Di Cambridge University di Inggris, antrian pekerjaan pada suatu waktu adalah garis pencucian (garis pakaian) dari mana kaset digantung dengan pakaian warna berbeda untuk menunjukkan prioritas pekerjaan.

3.1.7 Mikrokomputer

1. Mikrokomputer pertama tidak memiliki kapasitas atau kebutuhan untuk sistem operasi yang rumit yang telah dikembangkan untuk mainframe dan minis, sistem operasi minimalis dikembangkan, sering dimuat dari ROM dan dikenal sebagai monitor. Salah satu sistem operasi disk awal yang terkenal adalah CP / M, yang didukung oleh banyak mikrokomputer awal dan sangat ditiru oleh Microsoft MS-DOS, yang menjadi sangat populer sebagai sistem operasi yang dipilih untuk PC IBM. Pada 1980-an, Apple Computer Inc. meninggalkan seri mikrokomputer Apple II yang populer untuk memperkenalkan komputer Apple Macintosh dengan antarmuka pengguna grafis inovatif ke Mac OS.
2. Pengenalan chip CPU Intel 80386 pada bulan Oktober 1985, dengan arsitektur 32-bit dan kemampuan paging, menyediakan komputer pribadi dengan kemampuan untuk menjalankan sistem operasi multitasking seperti komputer minikomputer dan mainframe sebelumnya. Microsoft merespon perkembangan ini dengan mempekerjakan Dave Cutler, yang telah mengembangkan sistem operasi VMS untuk Digital Equipment Corporation. Dia dijadikan pemimpin dalam proses pengembangan sistem operasi Windows NT, yang terus-menerus berfungsi sebagai dasar untuk jalur/way sistem operasi Microsoft. Steve Jobs, co-founder Apple Inc., memulai NeXT Computer Inc., yang mengembangkan sistem operasi NEXTSTEP. NEXTSTEP nantinya akan diakuisisi oleh Apple Inc. dan digunakan, bersama dengan kode dari FreeBSD sebagai inti dari Mac OS X.
3. Pengenalan chip CPU Intel 80386 pada bulan Oktober 1985, dengan arsitektur 32-bit dan kemampuan paging, menyediakan komputer pribadi dengan kemam-

puan untuk menjalankan sistem operasi multitasking seperti komputer minikomputer dan mainframe sebelumnya. Microsoft merespon perkembangan ini dengan mempekerjakan Dave Cutler, yang telah mengembangkan sistem operasi VMS untuk Digital Equipment Corporation. Dia dijadikan pemimpin dalam proses pengembangan sistem operasi Windows NT, yang terus-menerus berfungsi sebagai dasar untuk jalur/way sistem operasi Microsoft. Steve Jobs, co-founder Apple Inc., memulai NeXT Computer Inc., yang mengembangkan sistem operasi NEXTSTEP. NEXTSTEP nantinya akan diakuisisi oleh Apple Inc. dan digunakan, bersama dengan kode dari FreeBSD sebagai inti dari Mac OS X.

4. Proyek GNU dimulai oleh aktivis dan programmer Richard Stallman dengan tujuan menciptakan penggantian perangkat lunak gratis yang lengkap ke sistem operasi UNIX yang berpaten. Meskipun proyek ini sangat berhasil dalam menduplikasi fungsionalitas berbagai bagian UNIX, pengembangan kernel GNU Hurd terbukti tidak produktif. Pada tahun 1991, mahasiswa ilmu komputer Finlandia Linus Torvalds, dengan kerja sama dari sukarelawan yang berkolaborasi melalui Internet, merilis versi pertama dari kernel Linux. Itu segera bergabung dengan komponen ruang pengguna GNU dan perangkat lunak sistem untuk membentuk sistem operasi yang lengkap. Sejak itu, kombinasi dari dua komponen utama biasanya hanya disebut Linux oleh industri perangkat lunak, konvensi penamaan yang Stallman dan Free Software Foundation tetap lawan, lebih memilih nama GNU / Linux. Berkeley Software Distribution, adalah bagian dari UNIX yang dirancang oleh University of California, Berkeley, dimulai pada thn 1970-an. Di distribusikan secara bebas dan diberikan ke banyak minikomputer, akhirnya mereka memperoleh pengikut untuk digunakan pada PC, terutama sebagai FreeBSD, NetBSD dan OpenBSD.

3.1.8 Berkeley Software Distribution

1. Subkelompok keluarga Unix adalah keluarga Distribusi Perangkat Lunak Berkeley, yang mencakup FreeBSD, NetBSD, dan OpenBSD. Sistem operasi ini paling sering ditemukan di webserver, meskipun mereka juga dapat berfungsi sebagai OS komputer pribadi. Internet berutang banyak Kebijakan untuk BSD, karena banyak dari protokol sekarang digunakan oleh komputer untuk menghubungkan, mengirim dan menerima data melalui jaringan diimplementasikan dan disempurnakan di BSD. World Wide Web Juga pertama kali dilakukan pada komputer yang menjalankan OS berdasarkan BSD yang disebut NeXTSTEP.
2. Pada tahun 1974, University of California, Berkeley menciptakan sistem Unix awal. Seiring waktu, mahasiswa dan staf di departemen ilmu komputer telah mulai menambahkan program baru untuk menyederhanakan, seperti editor teks. Ketika Berkeley menerima komputer VAX baru pada tahun 1978 dengan Unix-empat, para siswa di sekolah berbakat Unix lebih banyak menggunakan perangkat keras komputer. Departemen Pertahanan Advanced Defense Agency tertarik,

dan memutuskan untuk mendanai proyek tersebut. Banyak sekolah, perusahaan, dan organisasi pemerintah menggunakan versi Berkeley dari Unix dan bukan yang resmi yang digunakan oleh AT dan T.

3. Pada tahun 1974, University of California, Berkeley menciptakan sistem Unix awal. Seiring waktu, mahasiswa dan staf di departemen ilmu komputer telah mulai menambahkan program baru untuk menyederhanakan, seperti editor teks. Ketika Berkeley menerima komputer VAX baru pada tahun 1978 dengan Unix-empat, para siswa di sekolah berbakat Unix lebih banyak menggunakan perangkat keras komputer. Departemen Pertahanan Advanced Defense Agency tertarik, dan memutuskan untuk mendanai proyek tersebut. Banyak sekolah, perusahaan, dan organisasi pemerintah menggunakan versi Berkeley dari Unix dan bukan yang resmi yang digunakan oleh AT dan T.
4. Steve Jobs, setelah Apple Inc. pada tahun 1985, membentuk NeXT Inc., perusahaan yang memproduksi komputer high-end yang berjalan di bawah kondisi yang sama seperti BSD yang disebut NeXTSTEP. Salah satu komputer ini oleh Tim Berners-Lee sebagai webserver pertama yang menciptakan World Wide Web.
5. Pengembang seperti Keith Bostic mendorong proyek untuk mengurus kode non-bebas yang berasal dari Bell Labs. Setelah ini dilakukan, bagaimanapun, AT dan T menuntut. Setelah dua tahun sengketa hukum, proyek BSD menghasilkan beberapa derivatif gratis, seperti NetBSD dan FreeBSD (keduanya pada tahun 1993), dan OpenBSD (dari NetBSD pada tahun 1995).

3.1.9 Linux

1. Kernel Linux berasal pada tahun 1991, sebagai proyek Linus Torvalds, sementara seorang mahasiswa di Finlandia. Dia memposting informasi tentang proyek-proyek di newsgroup untuk siswa komputer dan programmer, dan Menerima dan membantu dari sukarelawan yang berhasil membuat kernel yang lengkap dan fungsional. Linux adalah Unix-like, tetapi dikembangkan tanpa kode Unix, tidak seperti BSD dan variannya. Karena model lisensi terbuka, kode kernel Linux tersedia untuk studi dan modifikasi, yang digunakan pada berbagai mesin dari superkomputer ke jam tangan pintar. Meskipun mereka menggunakan Linux pada 1,82

3.1.10 macOS

1. macOS (sebelumnya Mac OS X dan kemudian OS X) juga merupakan sistem operasi grafis inti yang dikembangkan, dipasarkan dan dijual oleh Apple Inc., yang terbaru yang telah dimuat sebelumnya pada semua komputer Macintosh yang sedang dikirimkan. macOS adalah penerus asli dari Mac OS klasik, yang telah menjadi sistem operasi utama Apple sejak 1984. Tidak seperti pendahulunya, ia telah dikembangkan di NeXT hingga tahun 1980-an dan sampai Apple

membeli perusahaan tersebut awal tahun 1997. Mac OS X Server 1.0, diikuti pada Maret 2001 oleh versi klien (Mac OS X v10.0 Cheetah). Sejak itu, enam klien dan edisi server yang lebih baik dari mac OS telah dirilis, untuk hal yang sama di OS X 10.7 Lion. Sebelum bergabung dengan macOS, edisi server - MacOS Server - adalah sama seperti mitra desktop dan juga berjalan di lini perangkat Apple Macintosh. MacOS Server menyertakan manajemen grup dan perangkat lunak yang mencakup akses ke layanan jaringan utama, termasuk transfer surat, server Samba, server LDAP, server nama domain, dan banyak lagi. Dengan Mac OS X v10.7 Lion, semua aspek server Mac OS X Server telah diintegrasikan ke dalam versi klien dan produk tersebut bermerek kembali sebagai OS X (menjatuhkan Mac dari namanya). Alat server sekarang tersedia sebagai aplikasi.

3.1.11 Microsoft Windows

1. Microsoft Windows adalah sistem operasi yang dirancang oleh Microsoft Corporation dan khususnya untuk komputer arsitektur Intel, dengan dimensi 88,9 persen dari total pada komputer yang terhubung dengan Web. Versi terbaru adalah Windows 10.
2. Pada 2011, Windows 7 mengambil alih Windows XP sebagai versi yang sangat umum. Microsoft Windows pertama kali dirilis pada tahun 1985, yang merupakan bagian dari MS-DOS, yang merupakan sistem operasi yang digunakan pada saat itu. Pada tahun 1995, Windows 95 dirilis hanya menggunakan MS-DOS sebagai bootstrap. Untuk menyelesaikan retret, Win9x dapat menjalankan driver MS-DOS dan Windows-3 Windows 16-bit yang nyata. Windows ME, dirilis pada tahun 2000, adalah versi terbaru dalam keluarga Win9x. Versi yang lebih baru sekarang didasarkan pada kernel Windows NT. Tanggung jawab Windows saat ini berjalan pada mikroprosesor ARM IA-32, x86-64, dan 32-bit. Selain itu Itanium masih didukung pada server lama versi Windows Server 2008 R2. Di masa lalu, Windows NT mendukung arsitektur tambahan. Server edisi Windows banyak digunakan.
3. Dalam beberapa tahun terakhir, Microsoft telah mengeluarkan modal yang signifikan dalam upayanya ke Windows sebagai sistem operasi server. Namun, penggunaan Windows pada server tidak meluas seperti pada komputer pribadi karena Windows bersaing dengan Linux dan BSD untuk server pasar. ReactOS adalah sistem operasi Windows alternatif, yang sedang dikembangkan pada prinsip-prinsip Windows - tanpa menggunakan kode Microsoft.

[?] [?] [?] [?] [?] [?] [?] [?] [?] [?]

Table 3.1 FAT

OS	Dukungan
MS-DOS	Ya
Windows 95	Ya
Windows 98	Ya
Windows Millenium Edition	Ya
Windows NT 3.x	Ya
Windows NT 4.0	Ya
Windows 2000	Ya
Windows XP	Ya
Windows Server 2003	Ya
Windows Vista	Ya

CHAPTER 4

OS OPEN SOURCE

4.1 Sistem Operasi Open Source

4.1.1 Definisi

Sistem Operasi Open Source yaitu sebuah sistem operasi yang source code dapat dibuka bebas oleh pengembangnya sehingga dapat dipelajari, diubah, dikembangkan, dan disebarluaskan lebih lanjut oleh setiap orang.

4.1.2 Sejarah Sistem Operasi Open Source

Open source pertama kali digagas oleh Eric S. Raymond, Christine Peterson, Todd Anderson, Larry Agustin, Jon Hall, dan Sam Ockman, yang dipimpin langsung oleh Richard Stallman pada tahun 1998. ini lah awal dari terbentuknya sistem operasi linux yang kita kenal saat ini.

CHAPTER 5

OS STARVATION

5.1 Starvation



Figure 5.1 Deadlock vs Starvation.

5.1.1 Definisi

Suatu kondisi yang biasanya terjadi setelah deadlock. Progres deadlock yang terjadi dapat mengakibatkan kekurangan resource, maka yang akan terjadi deadlock

Sistem Operasi, pre-release.

By Rolly Maulana Awangga Copyright © 2019 John Wiley & Sons, Inc.

tidak akan pernah mendapat resource yang dibutuhkan sehingga dapat mengakibatkan suatu kejadian, yaitu starvation atau kelaparan, Starvation berbarengan dengan deadlock sesuai dengan gambar 4.1. [?] Sistem computer dengan processor ganda memuat simpul pertama dan simpul kedua dipasangkan pada sebuah bus respon yang terpisah, dimana simpul pertama dan kedua berkomunikasi dengan mengirimkan permintaan paket pada bus respon terpisah. Ketika kesalahan terjadi dalam sebuah sistem, maka akan terjadi ketimpangan dalam pembagian sumber daya karena pada suatu proses selalu dibutuhkan pendapatan resource, sedangkan proses yang lain tidak pernah mendapatkannya starvation adalah kondisi yang biasanya terjadi setelah deadlock. Proses yang kekurangan sumber daya tidak akan mendapat sumberdaya yang dibutuhkan yang menjadikan sistem mengalami starvation . Starvation bisa terjadi tanpa deadlock saat terdapat kesalahan pada sistem sehingga terjadi ketimpangan dalam pembagian sumberdaya karena pada suatu proses selalu mendapat sebuah sumber daya, sedangkan proses yang lain tidak pernah mendapatkannya

5.1.2 Algoritma Starvation

Starvation dapat terjadi saat proses penjadwalan yang menggunakan prinsip proses yang lebih cepat akan diselesaikan terlebih dahulu, seperti pada Shortest Job First atau yang biasa di singkat SJF dan Penjadwalan Prioritas. Secara logis, misalkan individu mempunyai banyak sekali kebutuhan, maka individu tersebut akan memilih mana yang didahulukan berdasarkan sesuatu hal yang dibutuhkan.

5.2 Teknik menghadapi starvation

5.2.1 Menghindari Starvation

Beberapa cara dapat dilakukan saat menghadapi Starvation, salah satunya dengan Aging, proses awal yang ada diberi urutan (N) pemrosesan dengan rumus :

$$N = (P + T) / P .$$

N maksimum akan diproses dan proses lainnya akan dinaikkan tingkat urutan prosesnya supaya nanti saat ada proses lain yang masuk, proses sebelumnya akan mendapatkan bagian sumberdaya dan dapat diproses. Round Robin , adalah proses yang akan dimasukkan ke dalam antrian menurut proses kedatangannya. Suatu proses tidak akan selesai dengan cepat jika waktu yang dibutuhkan melebihi waktu yang sudah diberikan. Waktu kuantum itu sendiri adalah waktu yang telah diberikan untuk menyelesaikan suatu proses. Ketika proses telah mencapai batas waktu kuantum, sisa dari proses tersebut akan dikembalikan ke antrian terbawah dan sumber-sumber data akan dipindahkan ke proses selanjutnya. Maka dengan cara ini, semua proses yang mengantri akan mendapatkan sumber-sumber data secara bergantian (tidak ada proses yang memonopoli resource) sehingga semua proses dapat diselesaikan.

5.2.1.1 Menghambat Starvation dengan Disclosed [?] Disclosed adalah menghambat proses Starvation dalam Sistem Operasi multitasking dengan menyediakan

tipe pertama dari event penjadwalan pada interval waktu periodik, menyediakan tipe kedua dari event penjadwalan kedua sebagai tanggapan atas proses yang berjalan. Secara sukarela melepaskan prosesor dan, sebagai tanggapan atas acara penjadwalan menggantikan proses lama dengan yang baru, jika proses lama telah berjalan selama lebih dari satu jumlah waktu yang telah ditentukan. Sistem dijelaskan di sini menyediakan kernel kecil yang dapat dijalankan pada berbagai platform perangkat keras, seperti berbasis PowerPC Papan adaptor Symmetrix digunakan dalam Penyimpanan data Symmetrix perangkat yang disediakan oleh EMC Corporation of Hopkinton, Mass. Kode inti kernel dapat ditulis untuk target umum platform, seperti arsitektur PowerPC. Sejak Pow Modul spesifik implementasi erPC didefinisikan dengan baik, sistem mungkin cukup portabel antara prosesor PowerPC (seperti 8260 dan 750)

5.2.2 Ilustrasi

Bounded Waiting memiliki maksimum jumlah waktu yang diijinkan oleh proses lain untuk memasuki critical section atau disebut juga bagian penting setelah sebuah proses membuat permintaan untuk memasuki critical section-nya dan sebelum permintaan dikabulkan. Batasanbatasan itu biasanya menjamin proses dapat mengakses ke critical section (tidak mengalami starvation: proses se-olah berhenti menunggu request akses ke critical section diperbolehkan).

5.3 Menghambat Starvation dengan Disclosed

Disclosed adalah menghambat proses Starvation dalam Sistem Operasi multitasking dengan menyediakan tipe pertama dari event penjadwalan pada interval waktu periodik, menyediakan tipe kedua dari event penjadwalan kedua sebagai tanggapan atas proses yang berjalan. Secara sukarela melepaskan prosesor dan, sebagai tanggapan atas acara penjadwalan menggantikan proses lama dengan yang baru, jika proses lama telah berjalan selama lebih dari satu jumlah waktu yang telah ditentukan. Sistem dijelaskan di sini menyediakan kernel kecil yang dapat dijalankan pada berbagai platform perangkat keras, seperti berbasis PowerPC Papan adaptor Symmetrix digunakan dalam Penyimpanan data Symmetrix perangkat yang disediakan oleh EMC Corporation of Hopkinton, Mass. Kode inti kernel dapat ditulis untuk target umum platform, seperti arsitektur PowerPC. Sejak Pow Modul spesifik implementasi erPC didefinisikan dengan baik, sistem mungkin cukup portabel antara prosesor PowerPC (seperti 8260 dan 750)

5.4 Ilustrasi

- Terdapat tiga proses, yaitu Proses1, Proses2 dan Proses3.
- Proses1, Proses2 dan Proses3 memerlukan pengaksesan sumber daya R secara periodik Skenario berikut terjadi :

- Proses1 sedang diberi sumber daya R sedangkan Proses2 dan Proses3 diblocked menunggu sumber daya R.
- Ketika Proses1 keluar dari critical section, maka Proses2 dan Proses3 diijinkan mengakses R.
- Asumsi Proses3 diberi hak akses, kemudian setelah selesai, hak akses kembalidiberikan ke Proses1 yang saat itu kembali membutuhkan sumber daya R.
- Jika pemberian hak akses bergantian terus-menerus antara Proses pertama dan Proses ketiga, maka Proses kedua tidak akan mendapat akses sumber daya. Dalam kondisi ini memang tidak terjadi deadlock, hanya saja Proses kedua mengalami starvation atau tidak mendapat pelayanan yang dibutuhkan.

CHAPTER 6

OS DEADLOCK

6.1 DEADLOCK

6.1.1 Deadlock

6.1.1.1 Pengertian Deadlock Pada kesempatan ini saya akan menjelaskan tentang definisi Deadlock, Deadlock ialah suatu keadaan yang dimana dua proses atau lebih, saling menunggu proses untuk dapat melepaskan sumber daya yang sedang dijalankan. Misalnya proses A yang memerlukan suatu sumber daya, tetapi sumber daya tersebut sedang digunakan oleh proses lain. Untuk lebih paham mengenai pengertian dari deadlock dan bagaimana cara mengatasinya, anda dapat membandingkannya dengan situasi yang satu ini. Pertama, Dalam kehidupan kita tentu membutuhkan suatu pekerjaan, dan untuk memperoleh suatu pekerjaan, anda harus memiliki pengalaman yang baik, untuk dapat memiliki pengalaman yang baik anda harus bekerja.

Gambar 5.1 Contoh gambar pada saat terjadinya deadlock.

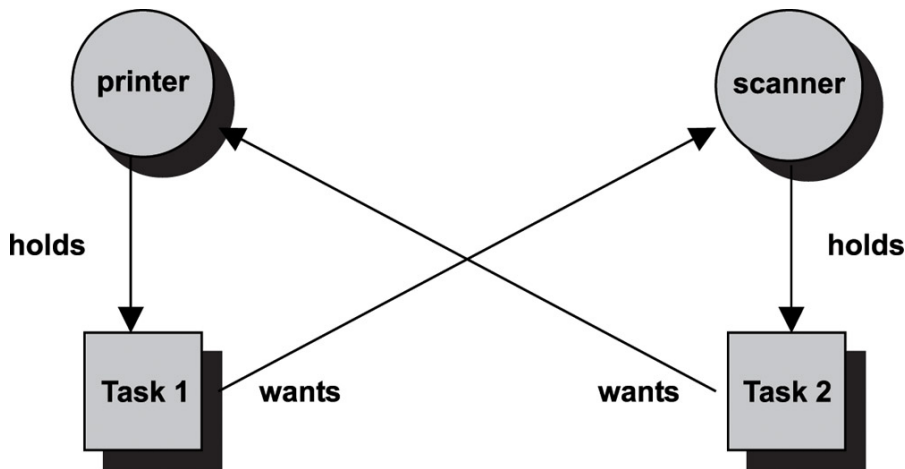


Figure 6.1 Gambar Deadlock

6.1.2 Masalah Deadlock dan Metode Penanganan Deadlock

6.1.2.1 Masalah Deadlock Deadlock merupakan dampak pengaruh dari sinkronisasi, yaitu dimana satu variabel yang digunakan oleh dua proses yang berbeda. Deadlock selalu tidak terlepas dari yang namanya sumber daya, karena hampir secara keseluruhan merupakan masalah mengenai sebuah sumber daya yang digunakan secara bersamaan. Sebuah Kelompok Proses yang diblok atau diblokir, dimana setiap proses memegang sebuah resource dan kemudian menunggu resource lain dari proses yang berada didalam proses yang sedang diBlok tersebut, biasanya dari semua proses-proses atau resource yang non preemptive..

6.1.2.2 Metode Penanganan Ada tiga Metode penanganan Deadlock: Yang Pertama yaitu, anda harus menggunakan satu protokol yang dapat membuat anda yakin bahwa sistem tersebut tidak akan pernah mengalami kejadian deadlock. Metode ini bisa disebut dengan Deadlock Prevention atau Avoidance.

Yang Kedua, anda harus memberikan izin sistem untuk mengalami kejadian deadlock, namun setelah terjadinya deadlock anda harus dengan cepat segera untuk memperbaiki sistem yang mengalami deadlock tersebut. Metode ini biasanya disebut dengan Deadlock detection and recovery.

Dan yang terakhir, anda hanya mengabaikan semua permasalahan yang terjadi secara bersamaan, dan kemudian menganggap bahwa deadlock tidak akan terjadi, metode ini digunakan dalam berbagai sistem operasi komputer, termasuk windows dan unix.

6.1.3 Deadlock Detection

1. 1. Pendeteksian secara Algoritma, yaitu dengan cara kita mengetahui jika terjadinya deadlock, deadlock terjadi jika suatu permintaan tidak dapat ditangani segera.
2. 2. Recovery atau Pemulihan, yaitu yang pertama menggagalkan semua proses deadlock, yang kedua mem backup semua proses yang deadlock dan kemudian silahkan melakukan restart di semua proses yang sedang terjadi, yang ketiga menggagalkan semua proses yang deadlock secara berurutan sehingga tidak akan terjadi lagi deadock, dan yang terakhir yaitu menggagalkan pengalokasian resource secara berurutan hingga tidak ada deadlock.

6.1.4 Beberapa hal yang terjadi ketika mendeteksi adanya deadlock

1. 1. Permintaan sumber daya dikabulkan selama memungkinkan.
2. 2. Sistem operasi melakukan scanning apakah ada kondisi circular wait secara peiodik.
3. 3. Pemeriksaan dilakukan setiap ada sumber daya yang hendak digunakan.
4. 4. memeriksa dengan algoritma tertentu.

6.1.5 Beberapa jalan untuk kembali dari deadlock

1. 1. Lewat Preemption, yaitu dengan jauhkan sumber daya dari pemakainya untuk sementara waktu, tujuannya untuk memberikannya pada proses lain. strategi dengan memberikannya kesempatan pada proses lain dengan tanpa diketahui oleh pemilik dari sumber daya itu dan tergantung juga dari sifat sumber daya itu sendiri.
2. 2. Lewat melacak kembali, setelah melakukan prosesn dari preemption tersebut maka secara otomatis proses utama yang diambil sumber dayanya akan stop dan tidak akan melanjutkan prosesnya, oleh karena itu dibutuhkan langkah untuk dapat kembali pada keadaan aman, tetapi untuk menentukan keadaan aman tersebut sangatlah susah.
3. 3. Mematikan proses yang menyebabkan deadlock, ini merupakan cara yang sangat umum digunakan yaitu dengan cara mematikan semua proses yang mengalami deadlock.
4. 4. Menghindari deadlock, pada sistem permintaan untuk sumberdaya biasanya hanya dilakukan sekali saja, sistem harus sudah dapat mengenali bahwa sistem itu aman atau tidak.

[?] [?] [?]

hline Proses	Jumlah Sumber Daya Digenggam	Maksimum Sumber Daya Dibutuhkan
X	2	10
Y	1	3
Z	3	7
Tersedia 4		

CHAPTER 7

OS

BAUD RATE

7.1 Baud Rate

7.1.1 Pengertian Baud Rate

Baud rate merupakan jumlah terjadi nya suatu perubahan status antara status nol dan status satu dalam waktu satu detik . Contoh nya ialah , suatu saluran memiliki sebuah baud rate 1000 , itu berarti saluran itu mampu mengirimkan setiap detik terjadi 1000 kali perubahan status .

7.1.2 Konsep Baud Rate

Komunikasi secara berturut-turut sudah tidak asing lagi di era teknologi ini, salah satunya dikarenakan jumlah penghantar yang digunakan bisa lebih efektif daripada melakukannya secara sejajar. Mengapa demikian? Karena kata Berturut-turut berarti mengirim satu bit data dan selanjutnya yang diikuti oleh bit-bit data yang lain pada jalur yang sama. Karena itulah kita dapat meringkas penggunaan kabel. Dikarenakan jalur yang dilalui bersamaan, maka kecepatan komunikasi berturut-turut tidak secepat kecepatan komunikasi sejajar. Komunikasi sejajar, dapat mengirim data se-

cara bersamaan melalui beberapa jalur. Namun, untuk proses secara keseluruhan, sistem komunikasi berturut-turut memenuhi berbagai aplikasi microcontroller. Selain itu, sistem komunikasi berturut-turut sering digunakan pada modem, USB, RS-232, dan teman-temannya. Hal yang sangat penting dalam menghubungkan dua perangkat melalui komunikasi berturut-turut adalah memastikan bahwa kedua perangkat berkomunikasi dengan bentuk yang sama, terdapat beberapa parameter yang digunakan untuk membangun komunikasi secara berturut-turut, diantaranya adalah Baud Rate, paket data, parity bit, dan synchronization bit.

Baud rate mengindikasikan seberapa cepat data dikirim melalui komunikasi berturut-turut. Satuan baud rate itu bit per second atau disingkat bps, walaupun untuk kasus tertentu dalam komunikasi sejajar, nilai bps bisa berbeda dengan nilai baud rate. Dugaan saat ini kita berpusat pada komunikasi berturut-turut, dimana setiap detik menyatakan transisi satu bit keadaan. Apabila hal ini terpenuhi, maka nilai baud rate akan sama dengan nilai bit per secondnya. Bit per second ini mengartikan bahwa berapa bit data dapat ditransfer setiap detiknya. Jika kita membalikan nilai bps ini, kita dapat memperoleh keterangan berapa lama waktu yang dipakai saat mengirim 1 bit. Nilai baud rate dapat diatur dengan standar kecepatan, diantaranya adalah sebagai berikut :

- 1.200bps
- 2.400bps
- 4.800bps
- 9.600bps
- 19.200bps
- 38.400bps
- 57.600bps
- 115.200bps

Kecepatan yang paling umum digunakan 9.600 bps. Ini adalah nilai yang mana kecepatan komunikasi bukanlah suatu hal yang kritis untuk dipertimbangkan. Seperti contoh, jika kita ingin mengetahui nilai dari sensor warna. Mendapatkan data warna dari suatu sensor tidaklah memerlukan kecepatan komunikasi yang terlalu cepat. Agar error tidak terjadi, kita menggunakan kecepatan standar 9.600 bps.

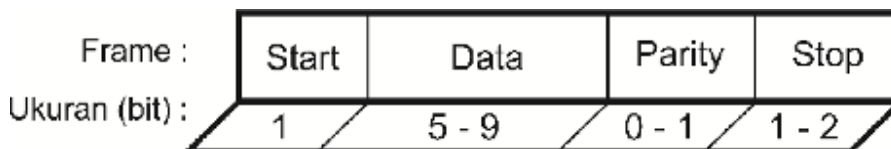


Figure 7.1 Serial Frame

7.1.3 Hubungan baud rate dengan bit rate

Baud rate yang tidak pernah bisa lebih tinggi dari channel bandwidth mentah, yang diukur dalam satuan Hz. Tingkat baudrate ini seringkali salah digunakan secara bergantian. Hubungan antara baud rate dan bit rate tergantung pada kecanggihan skema modulasi yang digunakan untuk memanipulasi carrier. Bit rate dan baud rate bisa sama jika setiap bit diwakili oleh transisi sinyal. Bit rate biasanya lebih tinggi sedangkan baud rate sebagai transisi sinyal tunggal dapat mewakili beberapa bit.

7.1.4 Fungsi Baud Rate

Baud rate dideteksi untuk mengetahui keluaran dari perangkat yang memerlukan basis serial untuk mengecek kecepatan data dan data itu sendiri. [?] Secara singkat, mendeteksi baud rate terdiri dari menentukan kecepatan transmisi dari perangkat yang mendapatkan sinyal karena perangkat penerima dapat dengan tepat mendecode sinyal dan mengkonversikannya ke perangkat. Baud rate memberitahu kecepatan data yang dapat dikirim melalui komunikasi serial. Dalam bps sendiri, berarti diketahui berapa kecepatan data yang dialirkan. Biasanya baud rate yang dipakai adalah 9600. Semakin besar baud rate yang dipakai, semakin tinggi kecepatan transfer data. Tetapi makin tinggi kecepatan maka makin beresiko mengalami error data. Untuk itu, disarankan untuk memakai baud rate standar atau dibawah 115.201.

7.1.5 Perbedaan Baud Rate Dengan Bit Rate

Kedua nya ini digunakan untuk mengukur kecepatan dalam konektivitas. Baud rate adalah ukuran berapa banyak simbol yang dikirimkan setiap sinyal. Simbol adalah setiap perubahan bentuk gelombang atau pulsa elektrik yang digunakan untuk mentransmisikan data sepanjang medium. Di sisi lain, bitrate adalah ukuran jumlah bit yang ditransmisikan. Dahulu, baud rate dan bit rate digunakan secara bergantian karena teknik modulasi yang lebih lama dan hanya memungkinkan satu bit untuk dimasukkan ke dalam setiap simbol. Namun sekarang, setiap simbol dapat berisi lebih dari bit, sehingga bitrate menjadi jauh lebih tinggi dari pada baud rate. Dengan kasus yang seperti ini, tidak lagi akurat untuk menggunakan bit rate dan baud rate secara bergantian karena tidak lagi sama. Bit rate kotor sama dengan baud rate dikali dengan jumlah bit yang terdapat pada masing-masing simbol. Karena baud rate tidak membedakan apakah informasi yang ditransmisikan adalah data atau hanya memberi isyarat informasi yang digunakan oleh perangkat keras untuk melakukan sinkronisasi, menentukan jalannya, dan banyak hal lain yang dilakukan pada sinkronisasi. Tingkat baud rate mencakup overhead. Jadi jika setiap simbol mengandung 4 bit, bit rate bersih sedikit kurang dari empat kali dari baud rate. Perbedaan nya biasanya sangat kecil dan seharusnya tidak mempengaruhi transmisi data yang terlalu banyak.

7.1.6 Framing Data

Framing data adalah teknik penyusunan data untuk dikirim melalui komunikasi serial. Pada gambar 6.1, Data yang dikirim melalui komunikasi serial biasanya dari 5 sampai 9 bit. Pada arduino, data yang dipakai berukuran 8 bit. Urutan dari pengiriman data biasanya mengikuti endian tertentu, seperti pengiriman most-significant-bit atau least-significant-bit terlebih dahulu yang dikirim. Framing data digunakan juga pada pengiriman data untuk mengframe data.

7.1.6.1 Kit Framing Data Start dan Stop bit biasa dikenal sebagai synchronization bit yang biasanya berukuran 2 atau 3 bit. Bit-bit ini mengawali dan mengakhiri pengiriman data. Start bit selalu diberi ukuran 1 bit sedangkan stop bit dapat diberi 1 atau 2 bit tergantung bagaimana pengguna perangkat keras atau pengkonfigurasi perangkat keras menggunakannya. Jika tidak memerlukan konfigurasi, nilai stop bit dapat dibiarkan menjadi sebesar 1 bit, seperti pada gambar 6.1. Posisi idle pada komunikasi serial bernilai 1. Start bit diidentifikasi dengan adanya transisi dari keadaan idle atau diam, yaitu dari 1 ke 0, sedangkan stop bit adalah kebalikan transisi dari keadaan idle untuk dari 0 ke 1. Bit Parity dalam framing data bersifat opsional dan dapat diabaikan. Parity bit digunakan untuk transfer data yang dipengaruhi oleh noise. Namun penggunaan bit parity dapat memperlambat kecepatan komunikasi atau transfer. Penggunaan bit parity juga dibutuhkan sebuah sinkronisasi antara transmitter dengan receiver karena dilakukan untuk mengurangi kemungkinan kesalahan dalam interpretasi data dalam perangkat keras.

7.1.7 Contoh Pengiriman Data

Dengan contoh saat kita mau mengirim kata OK, komunikasi akan memiliki 2 paket data. Pada Kode ASCII, untuk 'O' ialah 79 dalam bentuk desimal, sedangkan huruf 'K' yaitu 75. Data yang terkirim lebih dahulu adalah least-significant bit. Karena data dikirim dengan kecepatan 9600 bps, maka setiap bit memerlukan waktu selama $1/9600 = 104$ mikrodetik / bit. Dalam arduino, setting baud rate dapat dilakukan dengan kode :

```
Serial.begin(9600);
```

Kode tersebut menandakan bahwa kecepatan serial adalah 9600 bps atau 9600 bit per detik.

CHAPTER 8

OS SERIAL COMM

8.1 Komunikasi Serial pada Linux

8.1.1 Konsep Dasar Komunikasi Serial

Suatu komunikasi yang dilakukan dimana suatu pengiriman data dilakukan per bit ialah dinamakan komunikasi serial, sehingga akan lebih lambat jika dibandingkan dengan komunikasi parallel seperti yang ada pada port printer yang dapat mengirim 8 bit sekaligus dalam sekali detak. Terdapat 2 macam cara komunikasi data serial yaitu :

1. Komunikasi data serial sinkron
2. Komunikasi data serial asinkron

Terdapat 2 kelompok device pada komunikasi serial yaitu :

1. Data Communication Equipment (DCE) Contohnya seperti scanner, printer, modem dan yang lainnya.
2. Data Terminal Equipment (DTE) Contohnya seperti terminal yang ada pada komputer.

Keuntungan menggunakan port serial

- Masalah cable loss tidak akan menjadi suatu masalah yang besar pada komunikasi dengan kabel yang panjang, dari pada menggunakan kabel paralel. Port paralel akan mentransmisikan 0 volt dan 1 volt, sedangkan port serial akan mentransmisikan tegangan -3 - -25 volt dan tegangan +3 - +25 volt.
- Hanya membutuhkan jumlah kabel yang sedikit, menggunakan 3 kabel saja pun bisa yaitu saluran Ground, saluran Transmit Data, saluran Receive Data.
- Populernya penggunaan mikrokontroler dan kebanyakan mikrokontroler dilengkapi dengan Serial Communication Interface (SCI) yang bisa dipakai untuk melakukan komunikasi dengan port serial pada komputer.

8.1.2 Interprocess Communication

komunikasi antar proses untuk mengirim data dari satu proses ke proses yang lain, baik antar proses dalam satu komputer maupun proses dalam komputer yang berbeda karakteristik dari Interprocess Communication yaitu:

1. komunikasi Synchronous dan asynchronous pada Sinkronisasi Synchronous, proses pengiriman dan penerimaan pada setiap pesan dan sistem ini akan berfungsi, jika sistem mengirim pesan, maka sistem hanya akan dapat merespon, sampai pesan selesai. Dalam komunikasi asynchronous, komunikasi ini dapat langsung memproses pesan, begitu pesan berada di buffer lokal, dan mengirim pesan dengan benar.
2. Message destinations tempat tujuan dari sebuah pesan yang terdapat pada sebuah computer adalah local port, yang didefinisikan berarti sebagai variable angka dengan tipe integer. Sebuah port pasti mempunyai satu penerima, akan tetapi bisa memiliki banyak pengirim.
3. Reliability Keandalan suatu sistem dapat dilihat dari validitas dan integritas sistem. Sistem bila dilihat dari validitas, dapat dikatakan handal jika pesan yang disampaikan dijamin hingga tidak ada pesan yang hilang atau jatuh, dan sebaliknya.
4. Ordering menginginkan pesan yang terkirim dari pengirim dapat diterima sesuai dengan urutan grouping / ordering berdasarkan pesan awal yang terkirim.

8.1.3 Fungsi utama komunikasi serial

fungsi yang paling penting dari komunikasi serial pada mikrokontroler adalah untuk menyambungkan antara komputer dan mikrokontroler, dengan demikian kedua dapat bekerjasama.

8.1.4 contoh fungsi utama

contoh dari fungsi komunikasi serial seperti monitoring suhu menggunakan komputer, data suhu tersebut didapatkan oleh mikrokontroler lalu data tersebut ke komputer.

8.1.5 dua metode komunikasi serial

ada dua metode untuk komunikasi serial. pertama menggunakan port USB, metode ini disarankan karena USB terdapat pada PC yang dapat digunakan dimana saja pada usb port. kedua menggunakan menggunakan port serial, digunakan sebagai penghubung antara mikrokontroler dengan komputer PC.

8.1.6 macam-macam perintah terminal linux

macam macam perintah terminal pada linux:

1. cd (Change Directory) berguna untuk berpindah direktori. Sedangkan menggunakan perintah `cd` tanpa nama direktori akan mengantarkan anda kembali ke home direktori linux.
2. rm files yaitu berguna untuk melakukan penghapusan sebuah file.
3. mkdir berguna untuk membuat sebuah direktori yang baru, dengan membuat sebuah folder yang baru dengan menggunakan nama folder baru.

8.1.7 perbedaan port USB dan port serial

perbedaan port USB dengan port serial : perbedaannya adalah pada media penghubungannya, yaitu di port USB tidak bisa menggunakan rangkaian RS232 yang bisa digunakan sebagai media pengirim dan menerima data. Sedangkan pada metode port serial akan menggunakan sebuah rangkaian tambahan, yaitu sebuah rangkaian RS232 dan kabel serial yang digunakan sebagai penghubungnya.

8.1.8 cara menggunakan komunikasi serial menggunakan port serial menjadi port USB

cara komunikasi serial menggunakan port serial diubah menjadi port USB yaitu menggunakan sebuah alat konverter yang dinamakan serial to USB converter dan kabel USB male to male.

8.1.9 Koneksi Linux ke Serial Port

Untuk melakukan setting pada suatu perangkat, terkadang harus masuk terlebih dahulu ke dalam console box. Biasanya akan menggunakan hyperterminal, namun software bawaan seperti itu tidak terdapat pada linux pada saat linux terinstall. Maka

dari itu terdapat sebuah software yang dapat digunakan pada linux untuk melakukan komunikasi serial yaitu minicom untuk menggantikan hyperterminal.

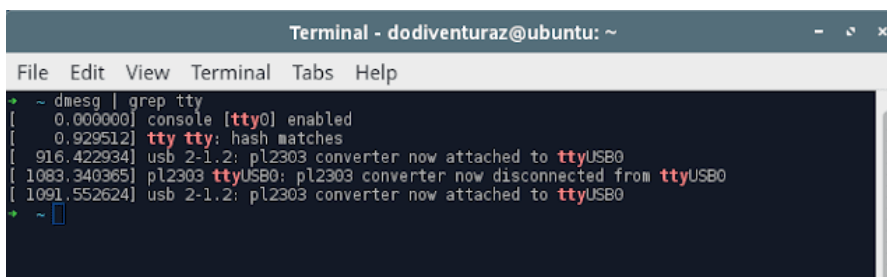
Software terminal minicom dapat di install dengan mudah di linux. Pertama buka terminal pada linux lalu ketik perintah :

`sudo apt-get install minicom`

Setelah itu software akan terinstall. Kemudian koneksikan ke perangkat yang akan digunakan menggunakan kabel console pada port serial. Lalu cek pada terminal linux dengan menggunakan perintah :

`dmesg — grep tty`

Perintah tersebut berguna untuk mengetahui port mana saja yang digunakan, seperti pada gambar dibawah ini



```

Terminal - dodiventuraz@ubuntu: ~
File Edit View Terminal Tabs Help
~ dmesg | grep tty
[ 0.000000] console [tty0] enabled
[ 0.929512] tty tty: hash matches
[ 916.422934] usb 2-1.2: pl2303 converter now attached to ttyUSB0
[ 1083.340365] pl2303 ttyUSB0: pl2303 converter now disconnected from ttyUSB0
[ 1091.552624] usb 2-1.2: pl2303 converter now attached to ttyUSB0
~ 

```

Figure 8.1 Gambar Status Serial

??

Lalu ketikkan perintah :

`sudo minicom -s`

dan akan muncul seperti dibawah ini :

Kemudian pilih Serial Port Setup untuk mengetahui port yang terdeteksi, lalu akan muncul tampilan sebagai berikut :

7.2

Lalu lakukan konfigurasi yang diperlukan sesuai kebutuhan yang diperlukan. Setelah konfigurasi selesai kembali ke menu utama dan pilih Save Setup as dfl. Kemudian pilih Exit, dan akan kembali ke terminal linux sebelumnya, kemudian ketikkan perintah

`minicom` Maka Perangkat akan terkoneksi sesuai keinginan.

Dirangkum dari makalah [?] Dirangkum dari makalah [?]



Figure 8.2 Gambar minicom

—configuration—
Filenames and paths
File transfer protocols
Serial port setup
Modem and dialing
Screen and keyboard
Save setup as dfl
Save setup as..
Exit
Exit from Minicom

CHAPTER 9

OS SERIALCOMWINDOWS

9.1 Serial Com Windows

9.1.1 Membuka Port

Dokumentasi SDK Platform menyatakan bahwa ketika membuka port komunikasi, panggilan ke CreateFile memiliki persyaratan berikut:

1. fdw Share Mode harus nol. Port komunikasi tidak dapat dibagikan dengan cara yang sama seperti file yang dibagikan. Aplikasi yang menggunakan TAPI dapat menggunakan fungsi TAPI untuk memfasilitasi berbagi sumber daya antar aplikasi. Untuk aplikasi yang tidak menggunakan TAPI, penanganan warisan atau duplikasi diperlukan untuk berbagi port komunikasi. Berurusan dengan duplikat berada di luar cakupan artikel ini, silakan merujuk ke dokumentasi Platform SDK untuk informasi lebih lanjut.
2. fdw Create harus menentukan bendera OPENEXISTING.
3. h Template File parameter harus NULL.

4. Satu hal yang perlu diperhatikan tentang nama port adalah bahwa mereka secara tradisional telah COM1, COM2, COM3, atau COM4. Windows API tidak menyediakan mekanisme apa pun untuk menentukan port apa yang ada pada sistem. Beberapa sistem bahkan memiliki lebih banyak port daripada maksimum tradisional empat. Vendor perangkat keras dan pembuat perangkat serial-driver bebas memberi nama port apa pun yang mereka sukai. Untuk alasan ini, yang terbaik adalah pengguna memiliki kemampuan untuk menentukan nama port yang ingin mereka gunakan. Jika port tidak ada, kesalahan akan terjadi setelah mencoba membuka port, dan pengguna harus diberitahu bahwa port tidak tersedia.
5. Satu hal yang perlu diperhatikan tentang nama port adalah bahwa mereka secara tradisional telah COM1, COM2, COM3, atau COM4. Windows API tidak menyediakan mekanisme apa pun untuk menentukan port apa yang ada pada sistem. Beberapa sistem bahkan memiliki lebih banyak port daripada maksimum tradisional empat. Vendor perangkat keras dan pembuat perangkat serial-driver bebas memberi nama port apa pun yang mereka sukai. Untuk alasan ini, yang terbaik adalah pengguna memiliki kemampuan untuk menentukan nama port yang ingin mereka gunakan. Jika port tidak ada, kesalahan akan terjadi setelah mencoba membuka port, dan pengguna harus diberitahu bahwa port tidak tersedia.

9.1.2 I atau O tumpang tindih

1. I atau O yang tumpang tindih tidak sesederhana I atau O non-tumpang tindih, tetapi memungkinkan lebih banyak fleksibilitas dan efisiensi. Sebuah port terbuka untuk operasi tumpang tindih memungkinkan beberapa utas untuk melakukan operasi I atau O pada saat yang bersamaan dan melakukan pekerjaan lain ketika operasi sedang menunggu. Lebih jauh lagi, perilaku operasi yang tumpang tindih memungkinkan satu utas untuk mengeluarkan banyak permintaan yang berbeda dan bekerja di latar belakang sementara operasi masih menunggu.
2. I atau O yang tumpang tindih tidak sesederhana I atau O non-tumpang tindih, tetapi memungkinkan lebih banyak fleksibilitas dan efisiensi. Sebuah port terbuka untuk operasi tumpang tindih memungkinkan beberapa utas untuk melakukan operasi I atau O pada saat yang bersamaan dan melakukan pekerjaan lain ketika operasi sedang menunggu. Lebih jauh lagi, perilaku operasi yang tumpang tindih memungkinkan satu utas untuk mengeluarkan banyak permintaan yang berbeda dan bekerja di latar belakang sementara operasi masih menunggu.
3. Baik dalam aplikasi single-threaded maupun multithread, beberapa sinkronisasi harus dilakukan antara mengeluarkan permintaan dan memproses hasilnya. Satu utas harus diblokir sampai hasil operasi tersedia. Keuntungannya adalah I atau O yang tumpang tindih memungkinkan utas untuk melakukan beberapa pekerjaan antara waktu permintaan dan penyelesaiannya. Jika tidak ada pekerjaan

yang dapat dilakukan, maka satu-satunya kasus untuk I atau O yang tumpang tindih adalah memungkinkan untuk respon pengguna yang lebih baik.

4. I atau O yang tumpang tindih adalah jenis operasi yang digunakan sampel MTTTY. Ini menciptakan sebuah thread yang bertanggung jawab untuk membaca data port dan membaca status port. Ini juga melakukan pekerjaan latar belakang secara berkala. Program ini menciptakan untaian lain secara eksklusif untuk menulis data di luar port.
5. Catatan Aplikasi terkadang menyalahgunakan sistem multithreading dengan membuat terlalu banyak utas. Meskipun menggunakan beberapa utas dapat menyelesaikan banyak masalah yang sulit, membuat untaian yang berlebihan bukanlah penggunaan yang paling efisien dalam aplikasi. Thread kurang regangan pada sistem daripada proses tetapi masih memerlukan sumber daya sistem seperti waktu CPU dan memori. Aplikasi yang menciptakan untaian berlebihan dapat mempengaruhi kinerja keseluruhan sistem.

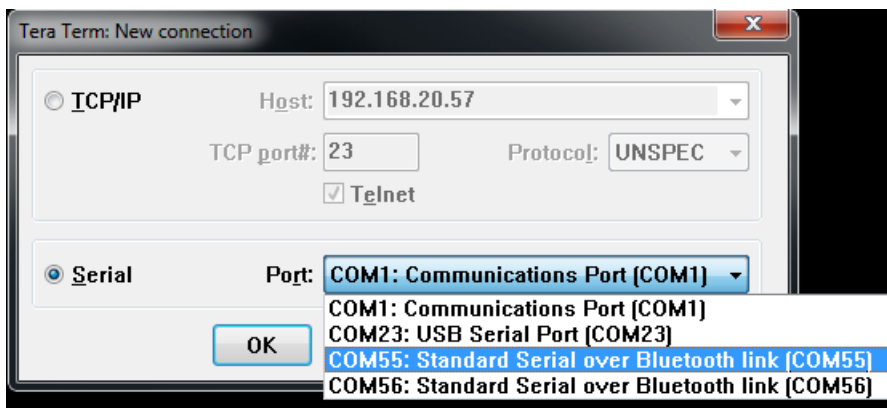


Figure 9.1 Serial Com Windows

Gambar 8.1 Contoh gambar.

```
HANDLE hComm;
hComm = CreateFile( gszPort,
                   GENERIC_READ | GENERIC_WRITE,
                   0,
                   0,
                   OPEN_EXISTING,
                   FILE_FLAG_OVERLAPPED,
                   0 );

if (hComm == INVALID_HANDLE_VALUE)
// error opening port; abort
```

9.1.3 Membaca dan menulis

1. Membaca dari dan menulis ke port komunikasi di Windows sangat mirip dengan file input atau output di Windows. Bahkan, fungsi yang melengkapi I atau O file adalah fungsi yang sama yang digunakan untuk serial I atau O. I atau O dapat dilakukan dengan salah satu dari dua cara: tumpang tindih atau tidak tumpang tindih. Dokumentasi SDK Platform menggunakan istilah asinkron dan sinkron untuk mengkonotasikan jenis operasi I atau O ini. Artikel ini, bagaimanapun, menggunakan istilah yang tumpang tindih dan tidak terabaikan.
2. Nonoverlapped I atau O akrab bagi kebanyakan pengembang karena ini adalah bentuk tradisional I atau O, di mana operasi diminta dan diasumsikan lengkap ketika fungsi kembali. Dalam kasus I atau O yang tumpang tindih, sistem dapat kembali ke pemanggil segera bahkan ketika operasi tidak selesai dan akan memberi sinyal kepada pemanggil ketika operasi selesai. Program ini dapat menggunakan waktu antara permintaan I atau O dan penyelesaiannya untuk melakukan beberapa pekerjaan latar belakang.

9.1.3.1 Bacaan

1. Fungsi ReadFile menerbitkan operasi baca. ReadFileEx juga mengeluarkan operasi baca, tetapi karena tidak tersedia pada Windows 95, itu tidak tercakup dalam artikel ini. Berikut adalah potongan kode yang merinci cara mempublikasikan permintaan baca. Perhatikan bahwa fungsi memanggil fungsi untuk memproses data jika ReadFile mengembalikan TRUE. Ini adalah fungsi yang sama yang disebut jika operasi menjadi tumpang tindih. Perhatikan flag fWaitingOnRead yang didefinisikan oleh kode; ini menunjukkan apakah operasi baca tumpang tindih atau tidak. Ini digunakan untuk mencegah penciptaan operasi baca baru jika mereka luar biasa.

```

DWORD dwRead;
BOOL fWaitingOnRead = FALSE;
OVERLAPPED osReader = {0};

// Create the overlapped event. Must be closed before exiting
// to avoid a handle leak.
osReader.hEvent = CreateEvent(NULL, TRUE, FALSE, NULL);

if (osReader.hEvent == NULL)
// Error creating overlapped event; abort.

if (!fWaitingOnRead) {
// Issue read operation.
if (!ReadFile(hComm, lpBuf, READ_BUF_SIZE, &dwRead, &osReader)) {
if (GetLastError() != ERROR_IO_PENDING) // read not delayed?
// Error in communications; report it.
```

```

else
fWaitingOnRead = TRUE;
}
else {
// read completed immediately
HandleASuccessfulRead(lpBuf, dwRead);
}
}

```

1. Bagian kedua dari operasi yang tumpang tindih adalah deteksi penyelesaiannya. Pegangan acara dalam struktur OVERLAPPED diteruskan ke fungsi WaitForSingleObject, yang akan menunggu hingga objek diberi isyarat. Setelah acara ditandai, operasi selesai. Ini tidak berarti bahwa itu berhasil diselesaikan, hanya saja itu selesai. Fungsi GetOverlappedResult melaporkan hasil operasi. Jika kesalahan terjadi, GetOverlappedResult mengembalikan FALSE dan GetLastError mengembalikan kode kesalahan. Jika operasi selesai dengan sukses, GetOverlappedResult akan mengembalikan TRUE.
2. Catatan Get Overlapped Result dapat mendeteksi penyelesaian operasi, serta mengembalikan status kegagalan operasi. Get Overlapped Result mengembalikan FALSE dan Get Last Error mengembalikan ketika operasi tidak selesai. Selain itu, Get Overlapped Result dapat dibuat untuk memblokir hingga operasi selesai. Ini secara efektif mengubah operasi yang tumpang tindih menjadi operasi non-tumpang tindih dan dicapai dengan melewati TRUE sebagai parameter bWait.

9.1.3.2 Penulisan

1. Pengarsipan data dari port komunikasi sangat mirip dengan membaca, karena menggunakan banyak API yang sama. Cuplikan kode di bawah ini menunjukkan cara menghapus dan menunggu operasi tulis selesai.

```

BOOL WriteABuffer(char * lpBuf, DWORD dwToWrite)
{
    OVERLAPPED osWrite = {0};
    DWORD dwWritten;
    DWORD dwRes;
    BOOL fRes;

    // Create this write operation's OVERLAPPED structure's hEvent.
    osWrite.hEvent = CreateEvent(NULL, TRUE, FALSE, NULL);
    if (osWrite.hEvent == NULL)
        // error creating overlapped event handle
        return FALSE;

    // Issue write.

```

```

if (!WriteFile(hComm, lpBuf, dwToWrite, &dwWritten, &osWrite)) {
    if (GetLastError() != ERROR_IO_PENDING) {
        // WriteFile failed, but isn't delayed. Report error and abort.
        fRes = FALSE;
    }
    else
        // Write is pending.
        dwRes = WaitForSingleObject(osWrite.hEvent, INFINITE);
    switch(dwRes)
    {
        // OVERLAPPED structure's event has been signaled.
        case WAIT_OBJECT_0:
            if (!GetOverlappedResult(hComm, &osWrite, &dwWritten, FALSE))
                fRes = FALSE;
            else
                // Write operation completed successfully.
                fRes = TRUE;
            break;

        default:
            // An error has occurred in WaitForSingleObject.
            // This usually indicates a problem with the
            // OVERLAPPED structure's event handle.
            fRes = FALSE;
            break;
    }
}
else
    // WriteFile completed immediately.
    fRes = TRUE;

CloseHandle(osWrite.hEvent);
return fRes;
}

```

9.1.4 Serial Status

1. Ada dua metode untuk mengambil status port komunikasi. Yang pertama adalah dengan mengatur event mask yang menyebabkan pemberitahuan aplikasi ketika peristiwa yang diinginkan terjadi. Fungsi Set Comm Mask mengatur masker kejadian ini, dan fungsi Wait Comm Event menunggu kejadian yang diinginkan terjadi. Metode kedua untuk mengambil status port komunikasi adalah secara berkala memanggil beberapa fungsi status yang berbeda. Polling, tentu saja, tidak efisien dan tidak direkomendasikan.

9.1.5 Communications Events

1. Komunikasi dapat terjadi kapan saja selama menggunakan port komunikasi. Dua langkah yang terlibat dalam menerima pemberitahuan acara komunikasi adalah sebagai berikut:
2. Set Comm Mask menetapkan peristiwa yang diinginkan yang menyebabkan pemberitahuan.
3. WaitCommEvent menerbitkan pemeriksaan status. Pemeriksaan status dapat berupa operasi tumpang-tindih atau non-tumpang tindih, seperti halnya operasi baca dan tulis.
4. Catatan : Peristiwa kata dalam konteks ini merujuk pada acara komunikasi saja. Itu tidak mengacu pada objek peristiwa yang digunakan untuk sinkronisasi.

```
DWORD dwStoredFlags;
```

```
dwStoredFlags = EV_BREAK | EV_CTS    | EV_DSR | EV_ERR | EV_RING | \
                EV_RLSD  | EV_RXCHAR | EV_RXFLAG | EV_TXEMPTY ;
if (!SetCommMask(hComm, dwStoredFlags))
    // error setting communications mask
```

9.1.6 Flow Control

1. Kontrol aliran dalam komunikasi serial menyediakan mekanisme untuk menangguhkan komunikasi sementara salah satu perangkat sibuk atau karena alasan tertentu tidak dapat melakukan komunikasi apa pun. Secara tradisional ada dua jenis kontrol aliran: perangkat keras dan perangkat lunak.
2. Masalah umum dengan komunikasi serial adalah operasi tulis yang sebenarnya tidak menulis data ke perangkat. Seringkali, masalah terletak pada kontrol aliran yang digunakan ketika program tidak menentukannya. Pemeriksaan dekat dari struktur DCB mengungkapkan bahwa satu atau lebih dari anggota berikut mungkin BENAR: fOutxCtsFlow, fOutxDsrFlow, atau fOutX. Mekanisme lain untuk mengungkapkan bahwa kontrol aliran diaktifkan adalah memanggil ClearCommError dan memeriksa struktur COMSTAT. Ini akan mengungkapkan ketika transmisi ditangguhkan karena kontrol aliran.
3. Sebelum membahas jenis-jenis pengendalian aliran, pemahaman yang baik tentang beberapa istilah sudah teratur. Komunikasi serial terjadi antara dua perangkat. Secara tradisional, ada PC dan modem atau printer. PC diberi label Data Terminal Equipment . DTE kadang-kadang disebut tuan rumah. Modem, printer, atau peralatan periferal lainnya diidentifikasi sebagai Peralatan Komunikasi Data . DCE kadang-kadang disebut sebagai perangkat.

[?] [?] [?]

