**LAPORAN PRAKTIKUM**

**SISTEM OPERASI RD**

**MODUL 3**

**Oleh :**

**Muhammad Yusuf (122140193)**



**Program Studi Teknik Informatika**

**Institut Teknologi Sumatera**

**2024**

# **Daftar Isi**

[**Daftar Isi** 2](#_Toc164802553)

[**1.** **Dasar Teori** 3](#_Toc164802554)

[**2.** **Ulasan Hasil & Jawaban** 3](#_Toc164802555)

[**3.** **Kesimpulan dan Saran** 10](#_Toc164802556)

# 

# **Dasar Teori**

Tulisan di atas membahas mengenai System Call, yang merupakan metode bagi program komputer untuk meminta layanan dari kernel sistem operasi di mana program tersebut dijalankan. Berikut adalah rangkumannya:

System Call (Panggilan Sistem)

1. Definisi: System call adalah metode program komputer untuk meminta layanan dari kernel sistem operasi.
2. Antarmuka: System call menyediakan antarmuka yang terdefinisi antara program pengguna dan sistem operasi melalui API.
3. Fungsi Utama:
4. Interface: Program membuat permintaan dengan memanggil fungsi tertentu, dan sistem operasi menjalankan layanan yang diminta.
5. Protection: System call digunakan untuk mengakses operasi istimewa yang tidak tersedia untuk program pengguna normal, dengan hak istimewa untuk melindungi sistem dari akses berbahaya atau tidak sah.
6. Kernel: Saat system call dibuat, program dialihkan dari mode pengguna ke mode kernel untuk mengakses sumber daya sistem.
7. Context Switching: System call memerlukan pengalihan konteks yang dapat menimbulkan biaya tambahan.
8. Error Handling: System call dapat mengembalikan kode kesalahan yang harus ditangani program dengan tepat.
9. Synchronization: System call digunakan untuk menyinkronkan akses ke sumber daya bersama.
10. Jenis System Call:
11. Process Control: Mengendalikan proses dalam sistem operasi.
12. File Management: Mengelola file dan direktori.
13. Device Management: Mengelola perangkat keras.
14. Information Maintenance: Mengelola informasi sistem.
15. Communication System: Mengelola komunikasi antar proses dan jaringan.
16. Memory Management: Mengelola memori dalam sistem operasi.

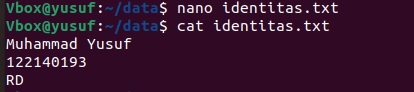
# **Ulasan Hasil & Jawaban**

**Percobaan 1 Open**

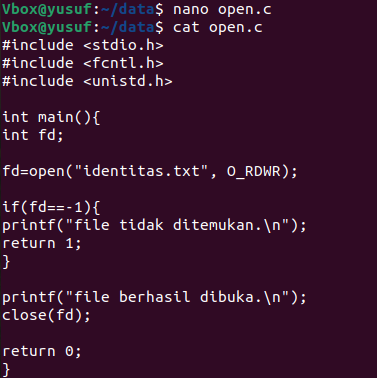
1. Membuat folder “Data” dan mengakses isi folder tersebut



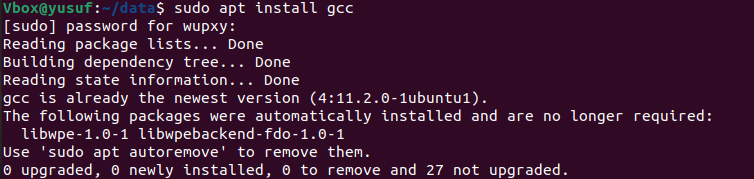
1. Membuat file dengan nama “identitas.txt” dengan isi Nama, Nim, Kelas



1. Membuaf file dengan ektensi C dengan nama “open.c”



1. Lakukan instalasi gcc dengan perintah “sudo apt install gcc” dan jalankan perintah “gcc open.c -o openfile”

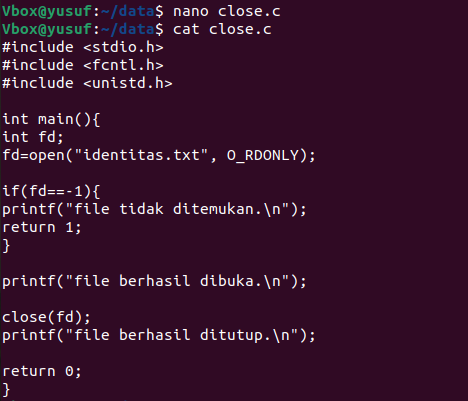


1. Lakukan pemanggilan sistem yang telah kita buat dengan perintah “./openfile”. Jika file berhasil dibuka maka akan menampilkan luaran berikut



**Perobaan 2 Close**

1. Mengakses folder “Data” dan membuaf file dengan ektensi C dengan nama “close.c”



1. Jalankan perintah “gcc close.c -o closefile”

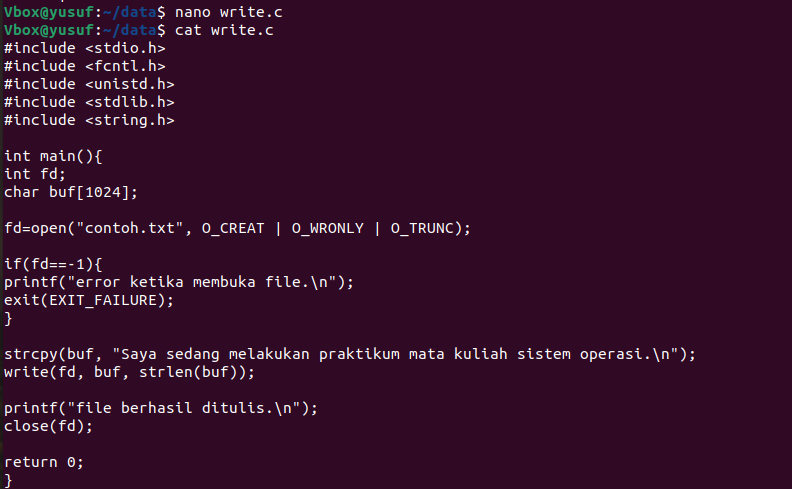


1. Lakukan pemanggilan sistem yang telah kita buat dengan perintah “./closefile”



**Percobaan 3 Write**

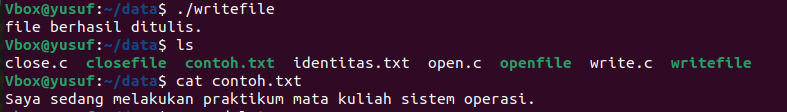
1. Mengakses folder “Data” dan membuaf file dengan ektensi C dengan nama “write.c”



1. Jalankan perintah “gcc write.c -o writefile”.

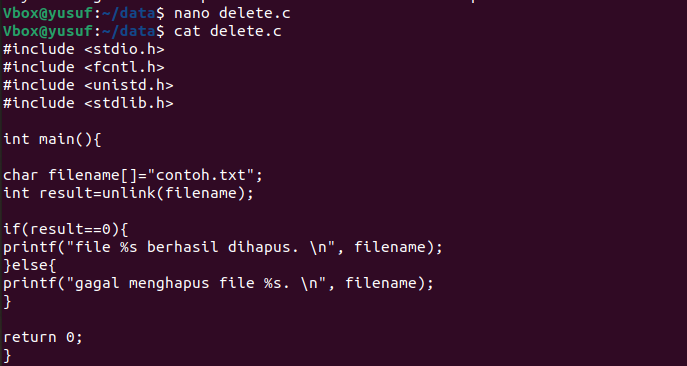


1. Lakukan pemanggilan sistem yang telah kita buat dengan perintah “./writefile”. Jika file berhasil ditulis maka akan menghasilkan file “contoh.txt”



**Percobaan 4 Delete**

1. Mengakses folder “Data” dan membuaf file dengan ektensi C dengan nama “delete.c”



1. Jalankan perintah “gcc delete.c -o deletefile”

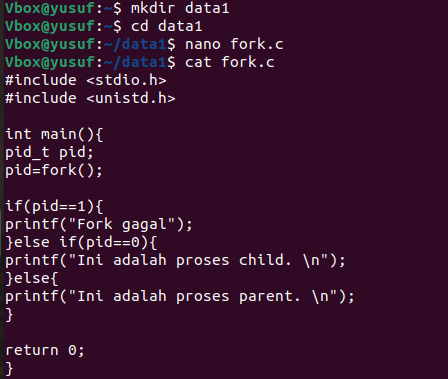


1. Lakukan pemanggilan sistem yang telah kita buat dengan perintah “./deletefile”. Jika file berhasil dihapus maka akan menghasilkan luaran



**Percobaan 5 Fork**

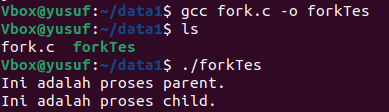
1. Mengakses folder “Data1” dan membuaf file dengan ektensi C dengan nama “fork.c”

****

1. Jalankan perintah “gcc fork.c -o ForkTes”

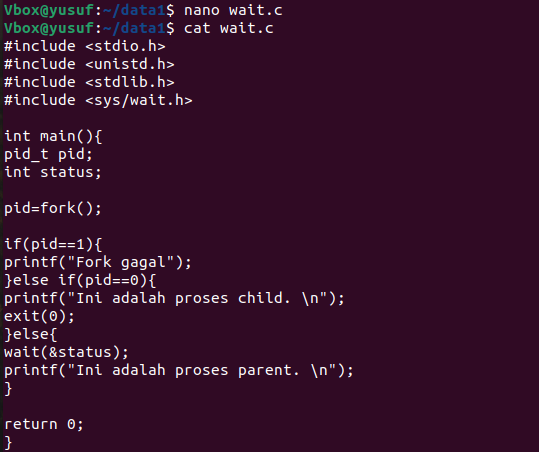


1. Lakukan pemanggilan sistem yang telah kita buat dengan perintah “./Fork.Test”. Jika file berhasil dihapus maka akan menghasilkan luaran

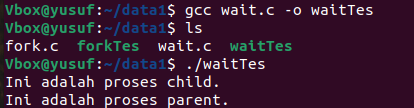


**Percobaan 6 Wait**

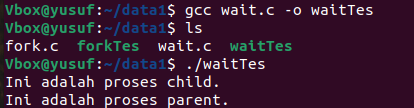
1. Mengakses folder “Data1” dan membuaf file dengan ektensi C dengan nama “wait.c”



1. Jalankan perintah “gcc wait.c -o WaitTes”



1. Lakukan pemanggilan sistem yang telah kita buat dengan perintah “./Wait.Test”. Jika file berhasil dihapus maka akan menghasilkan luaran



# **Kesimpulan dan Saran**

Dalam praktikum yang melibatkan pemahaman tentang System Call, saya dapat menarik beberapa kesimpulan kunci. Pertama, System Call merupakan jembatan krusial antara program pengguna dan kernel sistem operasi. Melalui panggilan sistem, program dapat meminta layanan tertentu dari sistem operasi seperti mengakses file, mengelola proses, mengelola perangkat keras, dan melakukan komunikasi antar proses. Antarmuka yang terdefinisi dengan baik memungkinkan program untuk berinteraksi dengan sistem operasi secara efisien dan aman.

Kedua, dalam penggunaan System Call, terdapat beberapa aspek yang perlu diperhatikan. Salah satunya adalah pentingnya perlindungan data dan hak istimewa yang diberikan kepada program melalui System Call. Selain itu, proses Context Switching saat melakukan panggilan sistem dapat memengaruhi kinerja sistem, sehingga diperlukan pengelolaan yang baik dalam meminimalkan biaya tambahan yang timbul. Kesalahan handling juga menjadi aspek penting yang harus diperhatikan agar program dapat merespons dengan tepat terhadap kondisi yang tidak diinginkan. Dengan pemahaman yang mendalam tentang System Call, program dapat berjalan dengan lebih efisien dan dapat diandalkan dalam mengakses sumber daya sistem.