

# 8

## **Sistem Berkas dan Direktori**



# Sistem Operasi

Departemen Teknik Informatika

Fakultas Ilmu Komputer, Universitas Brawijaya

Jl Veteran Malang 65145 Telp. (0341) 551611 ext: 126

---

## Modul 8

# Sistem Berkas dan Direktori

### 8.1. Tujuan Praktikum

Praktikum ini bertujuan memperkenalkan konsep system berkas dan direktori pada system operasi.

### 8.2. Capaian Praktikum

1. Mahasiswa mampu memahami konsep *file system* dan direktori.
2. Mahasiswa mampu menjelaskan konsep *file system* dan direktori dan menggunakan beragam perintah berkaitan kedua hal tersebut.

### 8.3. Dasar Teori

#### 8.3.1. File System

File System merupakan sebuah hal penting dalam sebuah sistem operasi. Mekanisme file system memberikan teknik bagaimana sebuah informasi diorganisasi dalam sebuah media penyimpanan. Sebuah file dapat dioperasikan dengan mekanisme sebagai berikut :

- a. File merupakan data yang bersifat abstrak
- b. Memiliki beberapa operasi seperti open, write, read, reposisi, delete, truncate, open(Fj), close (Fj) dan lain lain

#### 8.3.2. Direktori

Dalam file system Unix, sebuah **node indeks yang lebih dikenal dengan sebutan *inode***, yang merupakan struktur data struktur data yang digunakan untuk mewakili objek *filesystem*, yang dapat berupa **file atau direktori**. Setiap inode menyimpan atribut dan lokasi blok disk dari data objek filesystem. Atribut objek filesystem dapat mencakup manipulasi metadata. Metadata dari sebuah file dan direktori dapat berupa :

- a. Nama File
- b. Besar File
- c. Waktu Pembuatan, Perubahan, dan akses terakhir.



# Sistem Operasi

Departemen Teknik Informatika

Fakultas Ilmu Komputer, Universitas Brawijaya

Jl Veteran Malang 65145 Telp. (0341) 551611 ext: 126

- d. Permissions
- e. Filepath
- f. Checksum
- g. File data (inode)

Direktori hanya melakukan **pemetaan nama ke dalam inode**, POSIX dalam Unix menyediakan sebuah fungsi kecil untuk membaca nama file dan nomor inode untuk setiap entri. Dalam Unix, direktori dapat direpresentasikan sama dengan semua file dalam inode yang dapat dilustrasikan seperti pada gambar 7.1.

inode_num	name
2043567	hi.txt
...	

Gambar 8.1 Representasi Inode Number pada File dan Direktori

### 8.3.3. Hak Akses Direktori dan File

Setiap direktori dan file memiliki hak akses (permission) yang terdiri dari set bilangan biner sebanyak 9 bit. Yang dapat direpsentasikan sebagai :

- a. Hak akses read (3 bit)
- b. Hak akses write (3 bit)
- c. Hak akses execute (3 bit)

Selain hak akses, setiap file juga meiliki kepemilikan yang dapat diatur dan dipindahkan berdasarkan user atau group

## 8.4.Langkah Percobaan

Praktikum ini dilakukan dengan terlebih dahulu terhubung dengan layanan aws educate dengan cara mengaktifkan instance dari halaman instance summary. Pilih action dan Start untuk mengaktifkan instance. Lakukan koneksi SSH dengan cara yang sama seperti pada Bab 1.

### 8.4.1. Menulis File



# Sistem Operasi

Departemen Teknik Informatika

Fakultas Ilmu Komputer, Universitas Brawijaya

Jl Veteran Malang 65145 Telp. (0341) 551611 ext: 126

1. Login ke sistem GNU/Linux kemudian buka terminal.
2. Lakukan percobaan terhadap tiap kode program pada masing-masing sub bab berikut.
3. Lakukan kompilasi kode program dengan menggunakan perintah  
**gcc -o <Nama\_Output> <Nama\_Source.c>**
4. Jalankan hasil kompilasi menggunakan perintah **./<Nama\_Output>**
5. *Capture / Snapshot output*-nya dan simpan sebagai laporan.
6. Amati *output* tiap percobaan dan jawab pertanyaan pada masing-masing percobaan.

```
1#include <stdio.h>
2#include <stdlib.h>
3
4int main()
5{
6    FILE *fh;
7    fh = fopen("Praktikum_IX.txt", "w");
8    if (fh == NULL)
9    {
10        puts("Can't open that file!");
11        exit(1);
12    }
13    fprintf(fh, "Look what I made!\n");
14    fclose(fh);
15    return (0);
16}
```

## 8.4.2. Membaca File

1. Login ke sistem GNU/Linux kemudian buka terminal.
2. Lakukan percobaan terhadap tiap kode program pada masing-masing sub bab berikut.
3. Lakukan kompilasi kode program dengan menggunakan perintah  
**gcc -o <Nama\_Output> <Nama\_Source.c>**
4. Jalankan hasil kompilasi menggunakan perintah **./<Nama\_Output>**
5. *Capture / Snapshot output*-nya dan simpan sebagai laporan.
6. Amati *output* tiap percobaan dan jawab pertanyaan pada masing-masing percobaan



# Sistem Operasi

Departemen Teknik Informatika

Fakultas Ilmu Komputer, Universitas Brawijaya

Jl Veteran Malang 65145 Telp. (0341) 551611 ext: 126

```
1#include <stdio.h>
2#include <stdlib.h>
3
4int main()
5{
6    FILE *fh;
7    int ch;
8    fh = fopen("Praktikum_IX.txt", "r");
9    if (fh == NULL)
10    {
11        puts("Can't open that file!");
12        exit(1);
13    }
14    while ((ch = fgetc(fh)) != EOF)
15        putchar(ch);
16    fclose(fh);
17    return (0);
18}
```

7. Cupliklah hasil keluaran pada percobaan 7.4.1
8. Cupliklah hasil keluaran pada percobaan 7.4.2
9. Apa yang di maksud fh dan fopen (baris ke enam pada percobaan 7.4.1 dan baris ke tujuh percobaan 7.4.2)
10. Apa yang dimaksud dengan fungsi fclose dan fprint?

## 8.4.3. Membaca Direktori

1. Login ke sistem GNU/Linux kemudian buka terminal.
2. Buatlah sebuah direktori yang berisi beberapa (minimal 2) sub direktori. Isi sebuah file pada masing-masing
3. Lakukan percobaan terhadap tiap kode program pada masing-masing sub bab berikut.
4. Lakukan kompilasi kode program dengan menggunakan perintah  
**gcc -o <Nama\_Output> <Nama\_Source.c>**
5. Jalankan hasil kompilasi menggunakan perintah **./<Nama\_Output>**
6. Periksalah masing-masing keluaran dari perintah untuk setiap direktori



# Sistem Operasi

Departemen Teknik Informatika

Fakultas Ilmu Komputer, Universitas Brawijaya

Jl Veteran Malang 65145 Telp. (0341) 551611 ext: 126

7. *Capture / Snapshot output*-nya dan simpan sebagai laporan.
8. Amati *output* tiap percobaan dan jawab pertanyaan pada masing-masing percobaan

```
1#include <dirent.h>
2#include <stdio.h>
3#include <stdlib.h>
4
5int main(int argc, char **argv)
6{
7    if (argc == 1)
8    {
9        printf("Usage: %s [directory]\n", *argv);
10       exit(0);
11    }
12    struct dirent *dp;
13    DIR *dirp = opendir(argv[1]);
14    while ((dp = readdir(dirp)) != NULL)
15    {
16        printf("%s %lu\n", dp-> d_name, (unsigned long)dp-> d_ino
17    }
18    closedir(dirp);
19    return 0;
20};
```

9. Sebut dan jelaskan operasi direktori yang terdapat pada program diatas ?
10. Program yang berhasil anda kompilasi telah dimiliki oleh sistem Unix dan bahkan lebih kompleks, sebutkan beserta opsi dan keterangannya.

## 8.4.4. Percobaan mengatur hak akses (permission) file atau direktori

1. Login ke sistem GNU/Linux kemudian buka terminal.
2. Buatlah sebuah direktori atau file
3. Ketik perintah berikut secara bergantian

```
1$ chmod 644 <nama_file>
2$ chmod 755 <nama_file>
3$ chmod 700 <nama_file>
4$ chmod ugo-w <nama_file> $ chmod o-rx <nama_file>
```

4. Jalankan perintah `ls -l` dan cuplik tampilan outputnya



# Sistem Operasi

Departemen Teknik Informatika

Fakultas Ilmu Komputer, Universitas Brawijaya

Jl Veteran Malang 65145 Telp. (0341) 551611 ext: 126

---

## 8.4.5. Mengubah Kepemilikan File

1. Login ke sistem GNU/Linux kemudian buka terminal.
2. Buatlah sebuah direktori atau file
3. Buatlah beberapa user dengan perintah `adduser` (konsultasikan dengan asisten berkaitan dengan penggunaan perintah ini)
4. Ketik perintah berikut :

```
$ chown (user_baru) (nama_file)
```

5. Jalankan perintah `ls -l` dan cuplik tampilan outputnya
6. Jelaskan hak akses dari setiap file yang anda set permission-nya. Berapakah baset 9 digit hak aksesnya
7. Pada percobaan 7.4.4 cobalah ubah persimission file dengan perintah :  
  
**`chown root (nama_file)`**
8. Tampilkan keluarannya !
9. Bagaimanakah agar anda dapat mengubah sehingga file tersebut (pada tugas no 2) dapat berubah kepemilikan menjadi root
10. Tuliskan beberapa operasi file lainnya (minimal 4) , lakukan percobaan, cuplik dan tampilkan output lainnya

## 8.5. Kesimpulan

Jelaskan kesimpulan dari hasil percobaan pada Bab 8 ini

### UNTUK DIPERHATIKAN

Setiap selesai melakukan praktikum, jangan lupa menonaktifkan instance agar billing kuota tidak terus berjalan. Lakukan dengan cara memilih button Action pada Instance Summary. Pilih Stop sebagai perintah yang harus dijalankan.