Tugas Besar
IF2230 - Sistem Operasi
Milestone 02 of ??
“The Strength Within”
Pembuatan Sistem Operasi Sederhana
Folder dan Shell

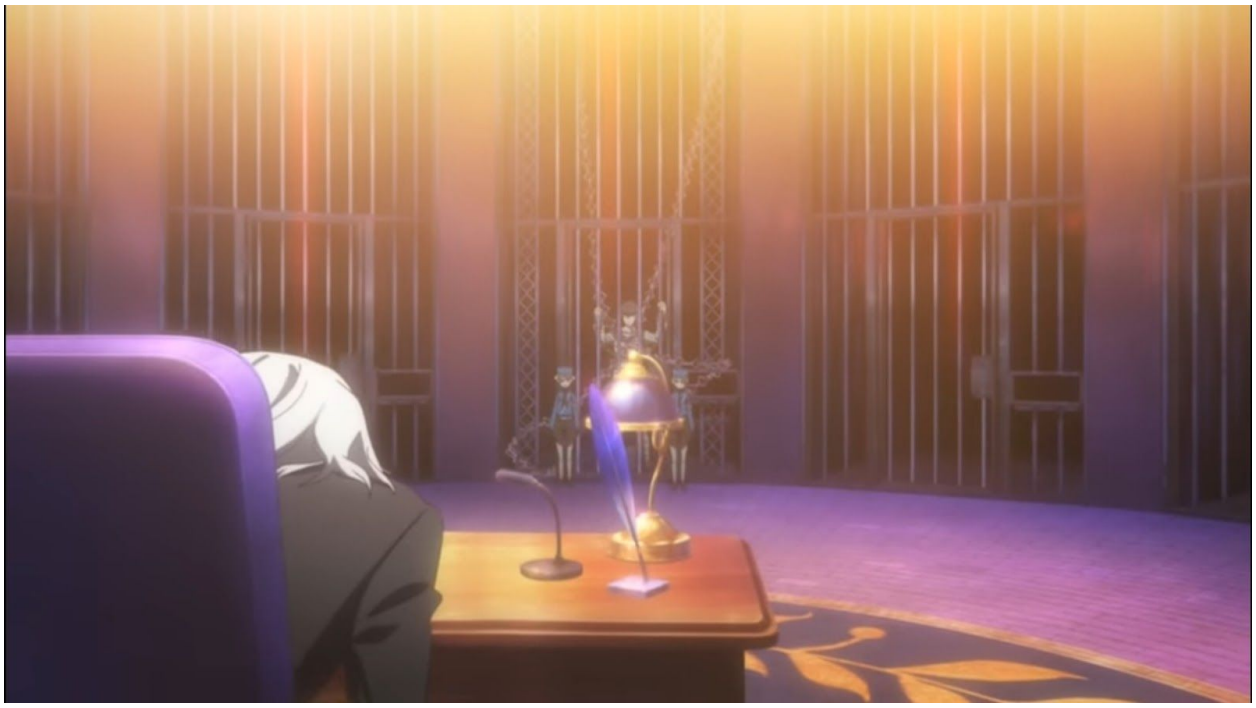
Dipersiapkan oleh :
Asisten Lab Sistem Terdistribusi

Didukung Oleh :

Waktu Mulai :
Rabu, 26 Februari 2020, 20.20 WIB

Waktu Akhir :
Jumat, 13 Maret 2020, 20.20 WIB

I. Latar Belakang



Setelah pengejaran yang cukup lama, akhirnya kalian berhasil melarikan diri dari istana tersebut. Kamu pun berhasil kembali ke dunia nyata, meskipun akhirnya kamu dimarahi oleh wali kelasmu karena datang terlambat.

Mulai hari itu, kamu pun mulai banyak menghabiskan waktu dengan teman barumu itu, yang kita sebut saja sebagai “Teman A”. Setelah beberapa lama *hangout* bersama, kalian pun memutuskan untuk membicarakan lagi apa yang terjadi di kastil itu beberapa hari yang lalu. Kalian pun akhirnya penasaran dengan apa yang terjadi di sana dan memutuskan untuk kembali ke sana pada esok harinya..

Pada malam harinya, kamu merasakan dirimu lebih lelah dari biasanya. Kamu pun memutuskan untuk secepatnya berbaring di tempat tidur. Tidak lama kemudian, kamu pun merasakan kesadaranmu jatuh ke tidur yang sangat lelap.

Ketika kamu mendapatkan kembali kesadaranmu, kamu membuka mata dan melihat ke sekelilingmu. Jauh dari apa yang kamu ingat mengenai kamarmu, kamu tidak melihat meja dan rak mu di mana pun. Di ruangan itu yang kamu lihat hanyalah kasur di mana kamu berbaring, yang entah kenapa tidak senyaman biasanya, dan sebuah jeruji layaknya apa yang di sebuah penjara.

Kamu mencoba mendekati jeruji tersebut untuk melihat keluar. Di hadapanmu berdiri dua orang gadis muda kembar dengan rambut pirang dan mata kuning yang berpakaian seperti dua orang sipir. Gadis yang satu memakai penutup mata di mata kiri dan gadis satunya lagi memakai penutup mata di mata kanan. Jauh di belakang mereka, terdapat seseorang yang

memiliki fitur yang sangat unik, yaitu hidung yang sangat panjang. Dia duduk di belakang sebuah meja. Dia pun menyambutmu.

Dia mengatakan bahwa kamu memiliki sebuah potensi langka di dalam dirimu. Sebuah potensi untuk menjadi lebih kuat lagi dari sebelumnya. Kuncinya adalah dengan menggabungkannya dengan “sesuatu” yang baru, “sesuatu” yang masih belum kamu miliki. Carilah hal tersebut, dan bangkitkan kekuatan sesungguhnya yang ada di dalam dirimu. Hanya dengan kekuatan itulah, kamu akan bisa mendapatkan apa yang kamu inginkan.

II. Deskripsi Tugas

Pada praktikum ini, kalian akan melanjutkan sistem operasi yang sudah kalian buat dengan menambahkan dukungan untuk folder dan sebuah *shell* sederhana

- Modifikasi *filesystem*
- Modifikasi *syscall* readFile dan writeFile
- Modifikasi loadFile
- Membuat sebuah *shell* sederhana

III. Langkah Pengerjaan

Semua instruksi berikut ini akan mengasumsikan Anda menjalankan Linux. Untuk program-program yang sudah harus terinstal pada sistem operasi Anda adalah sebagai berikut:

- Netwide assembler (https://www.nasm.us/) untuk kompilasi program assembly
- Bruce C Compiler (https://linux.die.net/man/1/bcc) untuk kompilasi C 16 bit (GCC sudah tidak mendukung 16 bit yang murni)
- ld86 (https://linux.die.net/man/1/ld86) untuk melakukan *linking object code*
- Bochs (http://bochs.sourceforge.net/) sebagai emulator untuk menjalankan sistem operasi

Pada Ubuntu 18.04 berikut adalah perintah yang dapat digunakan untuk menginstal program-program di atas

```
sudo apt install nasm bcc bin86 bochs bochs-x
```

3.1. Modifikasi *filesystem*

Pada *filesystem* kali ini akan ditambahkan dukungan untuk folder. Caranya adalah dengan menambahkan *flag* pada entri *file* untuk menandakan apakah sebuah entri adalah folder atau *file*. Selain itu kapasitas akan ditambahkan dengan menggunakan dua sektor untuk daftar *file* dan folder. Ditambah lagi, sektor yang digunakan akan disimpan di tempat berbeda agar menghemat tempat. *Flag* yang digunakan untuk menandakan folder juga berfungsi sebagai penanda indeks tabel sektor untuk *file*.

Jika bingung, bisa dilihat ilustrasi berikut ini

Struktur dari **2** sektor **files:**

idx	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	P	S	**Nama direktori/file 0** (14 karakter)													
01	P	S	**Nama direktori/file 1** (14 karakter)													
			...													
3F	P	S	**Nama direktori/file 63** (14 karakter)													

P : Indeks parent dari direktori tersebut. 0xFF jika parentnya root
S : Indeks entri pada tabel sektor (jika folder akan berisi 0xFF, jika file bisa berisi 0x00-0x1F)

Struktur dari sektor **sectors**:

idx	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	**Sektor *file* 0** (16 sektor)															
01	**Sektor *file* 1** (16 sektor)															
	...															
1F	**Sektor *file* n** (16 sektor)															

Jika dilihat, kita akan mempunyai 4 sektor total (termasuk map) untuk *filesystem*. Oleh karena itu, keempat sektor ini akan diletakkan pada posisi-posisi berikut:

- **map** (fungsi sama) di 0x100
- **files** di 0x101 dan 0x102
- **sectors** di 0x103

3.2. Modifikasi *syscall* readFile dan writeFile

Ubahlah *syscall* readFile dan writeFile untuk mendukung *filesystem* baru. Input filename menjadi path relatif ke sebuah file. *Path* akan relatif dari sebuah *parentIndex*.

```
void writeFile(char *buffer, char *path, int *sectors, char parentIndex);
void readFile(char *buffer, char *path, int *result, char parentIndex);
```

Agar fungsi *interrupt* dapat mendukung variabel >=4 maka AX dapat digunakan untuk 2 variabel seperti berikut

```
void handleInterrupt21 (int AX, int BX, int CX, int DX) {
   char AL, AH;
   AL = (char) (AX);
   AH = (char) (AX >> 8);
   switch (AL) {
      case 0x00:
         printString(BX);
         break;
      case 0x01:
         readString(BX);
         break;
      case 0x02:
         readSector(BX, CX);
         break;
      case 0x03:
         writeSector(BX, CX);
         break;
      case 0x04:
         readFile(BX, CX, DX, AH);
         break;
      case 0x05:
         writeFile(BX, CX, DX, AH);
         break;
      case 0x06:
         executeProgram(BX, CX, DX, AH);
         Break;
      default:
         printString("Invalid interrupt");
   }
}
```

Selain itu tambahkan sistem *error code* pada variabel success di readFile dan sectors di writeFile. Kode yang digunakan adalah sebagai berikut

Kode	**Arti (readFile)**	**Arti (writeFile)**
-1	File tidak ditemukan (readFile)	File sudah ada

-2		Tidak cukup entri di files
-3		Tidak cukup sektor kosong
-4		Folder tidak valid

3.3. Modifikasi loadFile

Modifikasilah loadFile.c sehingga mendukung *filesystem* yang baru. Diperbolehkan menggunakan bahasa pemrograman apapun.

3.4. Membuat *shell* sederhana

Agar tiap operasi di sistem operasi Anda menjadi lebih mudah, dibutuhkan sebuah *shell* sederhana. *Shell* ini akan memiliki fitur berikut

- Menampilkan *current directory* di *prompt*
- *Command history* dengan menekan tombol panah atas dan bawah (ukuran *history* minimal 3 perintah sebelum)
- Dukungan cd (*change directory*) secara relatif dengan "..". Misal: `cd ../../a/b/c/../d`
- Menjalankan program di folder yang sedang aktif. Misal: `./milestone2`
- Menjalankan program dari suatu folder tertentu. Misal: Ada folder bin di *root filesystem*, semua program di folder ini akan dapat dijalankan dari manapun dengan hanya mengetik nama program. Misal: `milestone2` (*Funfact*, pada sistem operasi umumnya, sistem ini biasa dinamai PATH, bisa dicoba dengan mengetik echo $PATH di terminal GNU/Linux untuk mengetahui lokasi folder-folder khusus ini)

3.4. Bonus

Bonus untuk milestone ini (diurutkan berdasarkan kesulitan dan skor)

- Adaptasi program kecil di milestone 1 agar kompatibel dengan sistem baru dan menambahkan penggunaan readFile dan/atau writeFile
- Autocomplete program yang tersedia
- Autocomplete folder saat melakukan cd
- Defragmenter

IV. Pengumpulan dan Deliverables

1. Untuk tugas ini Anda diwajibkan menggunakan *version control system* **git** dengan menggunakan sebuah *repository* ***private*** di github (gunakan surel *student* agar gratis)

2. Setiap kelompok akan diberikan *repository* dari asisten (yang nanti akan diberitahukan pembagiannya)

3. Walaupun *commit* tidak dinilai, namun diharapkan melakukan *commit* yang wajar (tidak semua kode satu *commit*)

4. File yang harus terdapat pada *repository* adalah file-file *source code* dan *script* (jika ada) sedemikian rupa sehingga jika diunduh dari github dapat dijalankan. Dihimbau untuk tidak memasukkan *binary* atau *image* hasil kompilasi ke *repository*

5. Isilkan nama kelompok dan anggotanya pada link berikut s.id/daftar-kelompok-os, paling lambat tanggal 5 Februari 2020.

6. **Mulai** Rabu, 26 Februari 2020, 20.20 WIB waktu server.
   Deadline Jumat, 13 Maret 2020, 20.20 WIB waktu server.
   Setelah lewat waktu *deadline*, perubahan kode akan dikenakan pengurangan nilai

7. Teknis pengumpulan adalah via kode yang terdapat di *repository* saat *deadline*

8. Kami akan menindaklanjuti **segala bentuk kecurangan**

9. Diharapkan untuk mengerjakan sendiri terlebih dahulu sebelum mencari sumber inspirasi lain (Google, maupun teman anda yang sudah bisa). Percayalah jika menemukan sendiri jawabannya akan merasa bangga dan senang.

10. Dilarang melakukan kecurangan lain yang merugikan peserta mata kuliah IF2230.

11. Jika ada pertanyaan atau masalah pengerjaan harap segera mengirimkan surel ke milis mata kuliah IF2230 Sistem Operasi.

V. Tips

1. Bcc tidak menyediakan *check* sebanyak gcc sehingga ada kemungkinan kode yang Anda buat berhasil *compile* tapi *error*. Untuk mengecek bisa mengcompile dahulu dengan gcc dan melihat apakah *error*
2. Untuk melihat isi dari *disk* bisa digunakan utilitas hexedit
3. Walaupun kerapihan tidak dinilai langsung, kode yang rapi akan sangat membantu saat *debugging*
4. Fungsi-fungsi dari *stdc* yang biasa Anda gunakan seperti mod, div, strlen, dan lainnya tidak tersedia di sini. Anda harus membuatnya sendiri, terutama mod dan div yang akan sangat berguna