

Tugas Besar

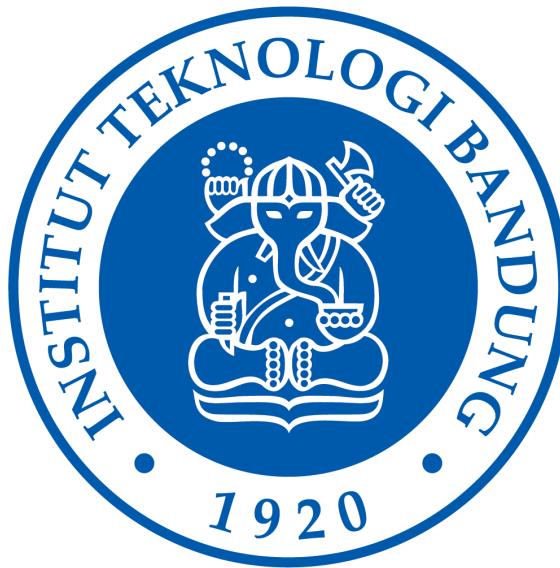
IF2230 - Sistem Operasi

Milestone 3 of 11

“Ho/OS”

**Pembuatan Sistem Operasi x86
Paging, User Mode, dan Shell**

Dipersiapkan oleh
Asisten Lab Sistem Terdistribusi



Waktu Mulai

Jumat, 31 Maret 2023, 18.00 WIB

Waktu Akhir

Sabtu, 29 April 2023, 23.59 WIB

I. Latar Belakang

Beberapa hari telah berlalu, perlahan namun pasti, *puzzle dunia bit* ini akan selesai secara keseluruhan. Mungkin.

Sementara itu, di bagian dunia lain, Pa Una yang dari kemarin-kemarin tidak kelihatan ternyata sedang melakukan hal lain. Pa Una memanfaatkan genjutsunya untuk menembus dimensi dunia bit dan sampai di suatu tempat rahasia. Sesuai ekspektasi Pa Una, dunia bit tersebut bekerja dengan prinsip yang sama dengan genjutsu miliknya. Pa Una melakukan ini bukan karena alasan, ia ingin bertemu langsung dengan *Gadis Misterius* itu untuk membahas sesuatu.. "Kita harus bicara, Omega."



"..." *Gadis Misterius*-untuk sekarang dan seterusnya akan dipanggil dengan nama aslinya, Omegaa-tidak terlihat tertarik memulai percakapan dengan Pa Una. Omegaa langsung menghilang dari tempat tersebut. "Menghindar? Jangan lari kamu!!". Pa Una kembali menggunakan genjutsu-nya untuk mengejar Omegaa. Tentu saja, Pa Una tidak akan melepaskan Omegaa begitu saja.

Sebuah permainan kejar-kejaran antara seorang gadis semi-tuhan dan gadis yang menyatu dengan alam, sebuah kombinasi yang menarik. Berlari kesana dan kemari melintasi ruang dan waktu, melompati dimensi, mengganggu keseimbangan dunia bit. Aksi kejar-kejaran

mereka secara tidak langsung merusak kerangka dunia *bit* tersebut. Meskipun demikian, tidak ada tanda-tanda dari kerusakan ini yang dapat kamu dan Docchi rasakan.

"Jangan jadi pengecut!!" Pa Una terus mengejar Omegaa, dan Omegaa terus menghindar darinya hingga akhirnya berhenti di suatu bagian di dunia *bit* yang cukup luas. Mereka menghentikan pergerakan, namun di antara mereka tidak ada tanda-tanda kelelahan yang terlihat. Mereka masih bisa kejar-kejaran hingga sekian millenium lagi.



Omegaa yang menolak Pa Una

AAAAAAAAAAAAA LUCUKKK

Stres benar anda - jek

"..." Omegaa masih menunjukkan ketidaktertarikannya untuk menjalin percakapan dengan Pa Una. Omegaa berdiri diam di sana, sementara Pa Una melangkah pelan mendekati Omegaa. Pa Una memiliki rasa penasaran yang diungkapkan dengan seribu dua pertanyaan di benaknya, yang siap ditumpahkan kepada Omegaa, mungkin menghanyutkannya jika Pa Una benar-benar terniat mengguyurnya dengan pertanyaan-pertanyaan tersebut. Bagaimana tidak, sejauh ini yang Omegaa sampaikan terkait kamu dan dunia *bit* ini adalah, bahwa kamu dapat menyelesaikan *puzzle* ini, dengan dibantu Pa Una, dan entah kenapa, Docchi, ~~jelas, karena sekarang Docchi Are~~, sisanya kamu hanya menyelesaikan tantangan dan diberikan, dan sedikit demi sedikit memahami bagaimana dunia *bit* ini bekerja, dan akhirnya, kamu akan mampu mengendalikan dunia *bit* ini.

Hm. "mampu mengendalikan dunia *bit*", dari sebuah *puzzle*, *a bit of a stretch, ain't it?* ~~Puzzle yang membuat kamu kehabisan waktu untuk mengerjakan tubes tubes lain~~ di desain sedemikian rupa agar orang-orang yang menyelesaikannya... dapat mengendalikannya? Agar

yang menyelesaikannya dapat menciptakan *puzzle* yang sama atau lebih baik? Atau sesuatu yang lebih besar lagi, seperti menguasai alam semesta? Atau yang lebih tidak masuk akal lagi, mengarungi dunia *multiverse* seperti di pilem dokter aneh dan membangun suatu jaringan yang berdiri di atas protokol dan kesepakatan multiversal? Hooooaaaoa????

"Aku hanya ingin tahu alasannya, Omega. Bagaimanapun juga, aku tidak akan mampu menghentikanmu kalau sudah sampai di sini. Bukan, aku tidak akan pernah bisa menghentikanmu dari awal. Memberitahukan alasannya saja tidak akan mengubah apa-apa, bukan?" Pa Una memohon dengan amat teramat sangat ~~seperti kamu yang memohon agar milestone 3 ini tidak lebih susah dari yang sebelumnya~~. Setelah beberapa saat, akhirnya Omegaa memalingkan kepalanya ke arah Pa Una yang berdiri tidak lebih dari 2 meter jaraknya dari Omegaa.

"Sebuah akhir... yang tidak dapat dihindari." Akhirnya Omegaa bersuara, meskipun yang dikatakannya tidak menjawab salah satu di antara seribu dua pertanyaan Pa Una, mungkin ada, tapi Pa Una tidak merasa demikian. Omegaa juga terlihat mengerutkan dahinya ketika mengatakan hal tersebut, seperti ingin menegaskan bahwa pernyataannya barusan memiliki makna yang signifikan.

"Akhir seperti apa? Akhir dari kehidupan? Ada kiamat? ~~Indeks jeblok?~~"

"Kamu tidak menyadarinya? Bahkan setelah kita sudah kehilangan satu?"

"Kita sudah move on dari masalah itu, bukan?!"

Nada Pa Una menaik, sebagian pertanyaan terjawab, tetapi Pa Una masih belum puas.

"Ck." Omegaa memalingkan wajahnya dari Pa Una, melihat ke arah lain yang sepertinya tertuju pada dua orang yang tengah berjalan ke arah yang menjauh. Dua orang itu adalah kamu dan Docchi, tetapi tidak ada di antara kalian yang menyadari sedang ditatap dari jauh.

"Semenjak ia 'tiada', Keseimbangan ruang dan waktu terus digangu oleh 'entitas' asing yang tidak kita ketahui. 'Warden of Time' kita tidak mampu memperbaikinya sendirian terus menerus. Sebelum kita menemukan pengganti dari yang sudah hilang, kita harus menyelesaikan masalah ini hingga ke akarnya".



Tidak terlalu singkat, tetapi padat dan cukup jelas. ~~Sebagian dari pembaca mungkin akan garuk-garuk kepala karena imajinasi penulis juga ikut meledak.~~ Pa Una hanya terdiam mendengar Omegaa yang tiba-tiba ngomong panjang. Satu per satu pertanyaan Pa Una terjawab, hingga menyisakan satu pertanyaan lagi. Pa Una melihat dua sosok yang juga dilihat Omegaa sebelumnya, dan sepertinya akan menjawab pertanyaan terakhir tersebut.

"...dan dia adalah orang yang menyelesaikan masalah tersebut?"

"..." Omegaa kembali tidak berkata apa-apa. Namun kali ini, tidak membuat Pa Una kesal. Pa Una paham karena inti dari *puzzle* ini adalah menjawab pertanyaan terakhir tersebut. Sekarang, Omegaa dan Pa Una hanya bisa menunggu. Tetapi tidak terlalu lama, karena sepertinya *puzzle* ini akan berakhir sebentar lagi. Dan setelahnya, pertanyaan tersebut akan terjawab dengan pasti.

Penulis mengakui bagian ini adalah *plot drop* yang amat sangat besar, saking besarnya penulis sendiri sempat bingung ketika melanjutkan cerita ini. Kesimpulannya, Omegaa sedang mencari seseorang yang mampu membantunya untuk menyelesaikan suatu masalah, dan Omegaa menciptakan dunia *bit* dan *puzzle* ini untuk mencari orang tersebut dan memastikan orang ini mampu dan sanggup menyelesaikan masalah tersebut. Masalahnya apa? Bisa dibilang, *Avengers' level threat*.

Kamu sepertinya telah menyelesaikan bagian kedua, dengan sekian banyak pengorbanan yang telah kamu lakukan, ~~seperti mengorbankan waktu tidur dan waktu santai seperti di~~

pantai. Kamu untuk yang sekian kalinya terduduk karena sudah sangat kelelahan menyelesaikan bagian sebelumnya.

"Ga masuk akal, nanti bakal gini lagi????? Bisa meledak aku."

Kamu maling untuk yang sekian kalinya. Sementara itu, Docchi yang melihat kamu mengmaling sedang memikirkan hal lain. Sesuatu yang lebih besar dan tidak mengenakkan. Mungkin itu hanya perasaan Docchi saja, namun perasaan tersebut telah mengganggunya beberapa hari belakangan. Docchi ingin menyampaikan perasaan tersebut ~~tukan-nembak ya, jangan geer~~ namun sepertinya ia tidak yakin.

Kamu melihat ke arah Docchi. Kamu menyadari kegelisahan Docchi yang terlihat dari ekspresi wajahnya. Docchi pun sedikit terkejut melihat kamu yang memandang serius ke arahnya. Docchi mencoba menghilangkan rasa gelisahnya dengan membalsas dengan senyuman, meyakinkan kamu kalau Docchi baik-baik saja, dan seharusnya memang terus seperti itu.

Kamu kembali mengarahkan fokus ke bagian selanjutnya dari *puzzle*. Kamu menyadari tingkatan abstraksi yang semakin tinggi dari *puzzle* ini. Kamu telah menemukan teknik mengendalikan bit-bit dan karena teknik tersebut sangat ribet dan pusing jika dilakukan dengan cara yang kamu pahami sekarang, kamu akan meledak. Meskipun demikian, bagian *puzzle* sebelumnya sudah memberikanmu *building blocks* untuk menyusun suatu sistem yang lebih mudah dan lebih baik.

Sesuatu yang menggabungkan teknik "menuliskan" perintah ke dunia bit, memanggil fungsi-fungsi untuk menciptakan suatu transformasi terhadap elemen-elemen dunia bit, dan struktur untuk menyimpan serangkaian bit tersebut.

Sebuah orkestrasi kendali kompleks. Sebuah *control room* yang sangat besar dan menyeluruh.

The Shell.

Kamu menyadari dunia bit ini sedikit lagi menjadi mirip seperti suatu komputer raksasa. Suatu robot Eva yang siap kamu piloti. Kamu akhirnya juga menyadari makna dari "mampu mengendalikan dunia bit" dari *puzzle* ini. Mungkin itu adalah petunjuk untuk penyelesaian *puzzle* ini secara menyeluruh.

Setelah mendapatkan ilham tersebut, kamu mengetahui apa yang selanjutnya harus lakukan. Kamu sekali lagi menggabungkan keseluruhan konsep informatika dan kamu pahami dengan semua bagian *puzzle* yang kamu kerjakan. Salah satu aspek penting dalam pembuatan

The Shell ini adalah keamanan yang harus dimaksimalkan. Kamu harus menciptakan cara agar *building blocks* yang sudah dibuat sebelumnya. Kamu mempelajari teknik **virtualisasi** yang dapat digunakan untuk mengamankan memori, dan kamu dapat membatasi akses dari *building blocks* tersebut dengan menciptakan **user mode**. Nantinya, **The Shell** akan memanfaatkan kedua hal tersebut.

Sangat rumit. Tetapi lagi-lagi, kamu memiliki kemampuan informatika yang sangat baik. Ini adalah momen yang sangat baik untuk memanfaatkan kemampuan tersebut. Setelah **The Shell** ini selesai, kamu dapat menciptakan fungsi-fungsi lain yang lebih kompleks, yang nantinya akan membuat keseluruhan dunia *bit* menjadi berada dalam kendali kamu, dan...

...terus apa? Apakah itu bagian akhir dari *puzzle*-nya? Kalau dipikir-pikir, Omegaa tidak pernah memberitahu bagaimana kondisi selesai dari *puzzle* ini. Kamu hanya memiliki petunjuk yang kamu simpulkan sendiri sebelumnya.

Berpikir hingga berlarut-larut tidak akan membawamu ke mana-mana. Saatnya kamu mengambil tindakan dengan menjalankan rencanamu membangun **The Shell**, senjata pamungkas untuk menaklukkan dunia ini, tentunya dengan bantuan Docchi.

"...skuy lanjut."

"skuy."

"...lebih cepat dari yang dibayangkan."

Demikian komentar Omegaa masih melihat kamu yang mulai mengerjakan bagian yang sepertinya merupakan bagian terakhir dari *puzzle*. Omegaa yang memandangmu dari kejauhan memuji performa kamu untuk yang kesekian kalinya. Pa Una tampaknya setuju dengan komentar Omegaa.

"Di antara sekian ide kamu, sepertinya kali ini kamu tidak salah, Omega." Pa Una menanggapi komentar Omegaa sebelumnya, sekaligus menyindirnya dengan mengungkit sedikit masa lalu yang sepertinya penuh dengan percobaan yang berujung kegagalan. Omegaa tidak mengatakan apa-apa lagi setelahnya.

"Ngomong-ngomong, gadis yang di dekatnya itu siapa? Kenapa ada di sini?" tanya Pa Una.

"Mau tau aja."

II. Deskripsi Tugas

Milestone ini adalah “**penggunaan milestone sebelumnya**”. Masih sama dan melanjutkan dengan milestone sebelumnya, milestone ini akan melanjutkan penggerjaan dari milestone 2 (File System).



Post M2 : Dead and Exhausted

Meskipun dari kedua milestone sebelumnya sudah menyediakan basic I/O, kedua milestone tidak menggunakan I/O tersebut untuk hal yang berguna. Milestone ketiga akan berfokus kepada **Shell** yang berjalan pada **User Mode**. Untuk mencapai Shell, diperlukan beberapa tahapan seperti **Paging** dan **User Mode preparation** terlebih dahulu. Berikut adalah garis besar yang akan dikerjakan pada milestone 3

- Paging
- User Mode
- Shell

File-file template dan contoh milestone 3 dapat diakses pada repository berikut

[Sister20/kit-OS-milestone-3-2023](#)

Guidebook tugas besar untuk milestone 3 dapat diakses pada link berikut

 IF2230 - Guidebook - Milestone 3

Catatan penting: Jika masih ada permasalahan / mengalami stuck dengan milestone 1 atau 2, resolve secepatnya dan gunakan asistensi jika perlu

III. Spesifikasi Tugas

3.0. Outline Pengerjaan

Sama seperti milestone 2, dokumen ini akan berisikan requirement yang harus dikerjakan pada milestone 3. Penjelasan yang lebih detail akan dipisahkan ke dokumen guidebook.

Pengerjaan milestone 3 dapat dipisah menjadi *appetizer* dan *main dish*

Appetizer

- **3.1. Paging**
 - Paging Data Structures
 - Higher Half Kernel
 - Activate Paging
- **3.2. User Mode**
 - Inserter
 - User GDT & Task State Segment
 - Simple Memory Allocator
 - Simple User Program
 - Launching User Mode

Main dish

- **3.3. Shell**
 - System Calls
 - Shell implementation
 - Built-in Utility Command

3.1. Paging

User program nantinya akan berjalan pada virtual memory yang memiliki privilege user. Tentunya sebelum membuat user program ini, sistem operasi perlu menyediakan virtual memory terlebih dahulu. Bagian ini akan mengimplementasikan Paging yang nantinya akan digunakan untuk virtual memory user program

3.1.1. Paging Data Structures

Paging membutuhkan beberapa struktur data untuk menyimpan informasi pemetaan virtual memory. Buatlah deklarasi struktur data paging pada file **paging.h**:

- **PageDirectoryEntryFlag**
- **PageDirectoryEntry**
- **PageDirectory**

3.1.2. Higher Half Kernel

Tugas besar ini akan menggunakan layout virtual memory **Higher Half Kernel**.

- Ganti relocation address kernel ke address **0xC010 0000** tetapi dengan load address **0x0010 0000**.

3.1.3. Activate Paging

Setelah relocation address kernel sudah disetup dan struktur data paging sudah disetup, nyalakan paging sistem operasi.

- Nyalakan paging pada sistem operasi menggunakan kode assembly

3.2. User Mode

Shell nantinya adalah sebuah program yang berjalan pada user mode. Sebelum memasuki shell, diperlukan beberapa persiapan untuk memasuki user mode.

3.2.1. External Program - Inserter

Program user akan di-compile secara terpisah dari kernel. Hasil executable juga terpisah dari kernel dan nantinya akan dimasukkan pada file system sistem operasi.

- Buatlah sebuah **program eksternal / pada host OS** yang dapat memasukkan file ke dalam storage image sistem operasi.

3.2.2. User GDT & Task Segment State

GDT yang dibuat sebelumnya hanya akan digunakan untuk **privilege level**, fitur segmentation akan diabaikan. Nantinya pada system calls, diperlukan inter privilege interrupt yang membutuhkan **Task Segment State** yang sudah disetup ke CPU.

- Buatlah 3 entry pada GDT yang mencakup user GDT dan TSS descriptor.
- Definisikan juga satu TSS yang akan digunakan pada TSS descriptor.

3.2.3. Simple Memory Allocator

Sistem paging yang dibuat pada bagian sebelumnya masih belum memiliki cara untuk mengalokasikan virtual memory ke physical memory.

- Lanjutkan implementasi **allocate_single_user_page_frame()** yang ada pada **paging.c**

3.2.4. Simple User Program

Setelah hampir semua komponen user mode sudah siap, salah satu hal yang penting belum diimplementasikan adalah program itu sendiri

- Implementasikan user program sederhana dengan format **flat binary executable** dan **insert ke dalam file system** menggunakan inserter yang telah dibuat

3.2.5. Execute Program

Langkah terakhir sebelum mencoba untuk mengeksekusi user program adalah membuat “jump” ke instruksi user program.

- Buatlah fungsi `kernel_execute_user_program()` yang mengeksekusi user program dan `set_tss_register()` untuk set TSS register CPU dengan value yang benar

3.2.6. Launching User Mode

Setelah semua step sebelumnya telah dibuat, uji hasil implementasi dengan kode berikut

```
kernel.c

void kernel_setup(void) {
    enter_protected_mode(&_gdt_gdtr);
    pic_remap();
    initialize_idt();
    activate_keyboard_interrupt();
    framebuffer_clear();
    framebuffer_set_cursor(0, 0);
    initialize_filesystem_fat32();
    gdt_install_tss();
    set_tss_register();

    // Allocate first 4 MiB virtual memory
    allocate_single_user_page_frame((uint8_t*) 0);

    // Write shell into memory (assuming shell is less than 1 MiB)
    struct FAT32DriverRequest request = {
        .buf            = (uint8_t*) 0,
        .name           = "shell",
        .ext            = "\0\0\0",
        .parent_cluster_number = ROOT_CLUSTER_NUMBER,
        .buffer_size     = 0x100000,
    };
    read(request);

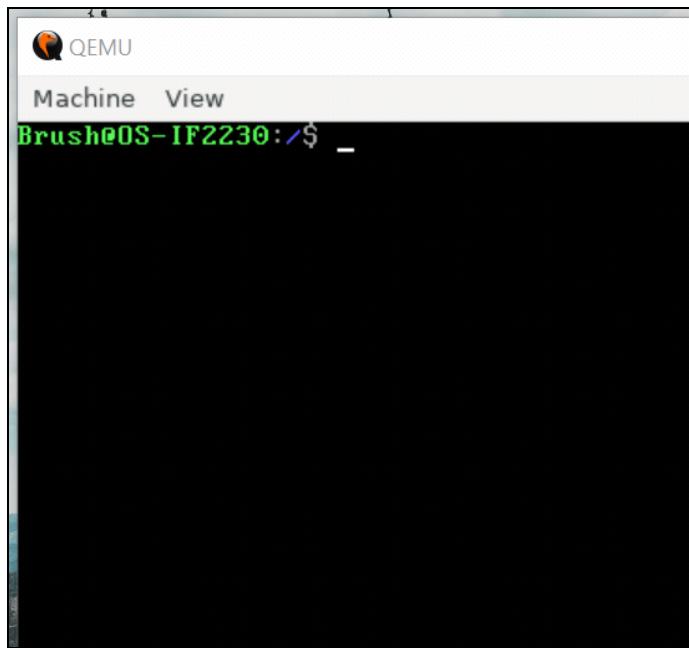
    // Set TSS $esp pointer and jump into shell
    set_tss_kernel_current_stack();
    kernel_execute_user_program((uint8_t *) 0);

    while (TRUE);
}
```

Jika berhasil, QEMU akan diam dan memiliki register `eip` sekitar 0x0 hingga 0x100

3.3. Shell

Setelah semua bagian sebelumnya sudah diimplementasikan dengan baik, maka **end game** sudah dimulai



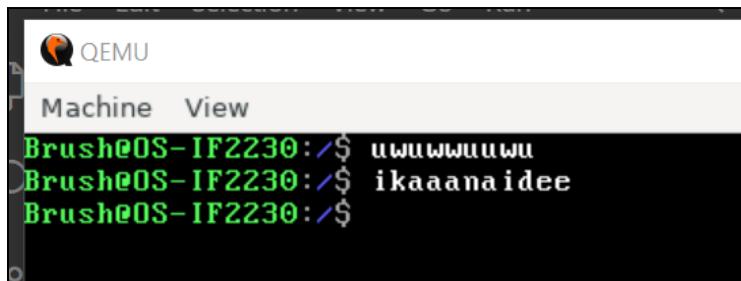
Shell yang dibuat merupakan user program yang memiliki built-in utility commands (**bukan executable terpisah**).

3.3.1. System Calls

Untuk bagian ini, implementasi system calls dibebaskan. Hanya saja, ganti **initialize_idt()** yang dibuat pada milestone 2 untuk menggunakan privilege level 0x3 untuk beberapa **InterruptGate**. Interrupt memiliki privilege 3 akan dapat diakses oleh user program dan menjadi cara utama untuk melakukan system calls.

- Implementasikan system calls sesuai kebutuhan shell
- *Sebisanya untuk mengimplementasikan System Calls sesuai dengan UNIX dan lain-lain*
- **Opsional** : Implementasikan standard library (untuk sisi user) seperti **string.h**, **stdio.h**, dll

3.3.2. Shell Implementation



Implementasi shell akan sebagian besar diserahkan kepada praktikan. Namun ada beberapa requirement yang perlu dipenuhi untuk penilaian.

Berikut adalah behavior dasar yang harus ada:

- Shell akan menggunakan desain **REPL** (read-eval-print-loop, seperti shell biasanya)
- Shell dapat menuliskan **Current Working Directory** (CWD)
- Awal current working directory terletak pada **Root**
- **Shell tidak wajib mem-parse relative pathing**, hanya absolute dan di-dalam cwd saja
(Contoh *relative pathing* yang tidak perlu di-*implement*: `cd ./f1` atau `cd ./f1/../f1`)

Spesifikasi meminta *built-in* utility berikut pada shell, dengan behavior UNIX-like (selain **whereis**):

- **cd** - Mengganti current working directory (termasuk .. untuk naik)
- **ls** - Menuliskan isi current working directory
- **mkdir** - Membuat sebuah folder kosong baru
- **cat** - Menuliskan sebuah file sebagai text file ke layar (Gunakan format LF newline)
- **cp** - Mengcopy suatu **file**
- **rm** - Menghapus suatu **file atau folder kosong**
- **mv** - Memindah dan merename lokasi file/folder
- **whereis** - Mencari **file/folder dengan nama yang sama diseluruh file system**

3.4. Bonus

- Recursive **cp & rm (5; 2.5 per command)**

Tambahkan fitur untuk menghapus/copy sebuah direktori dengan flag tambahan **-r**. **Flag ini akan secara rekursif menghapus/copy** isi dari sebuah direktori. Penghapusan metadata parent folder saja tidak dihitung mengerjakan bonus ini. **Mind the call stack usage**

- Relative pathing **(2.5)**

Shell mendukung **relative pathing** untuk argumen command. Contohnya **./f1/..../f2** atau **../f1/f2/f3**

- Indexing **(7.5)**

Command **whereis** membutuhkan pengecekan secara rekursif dari root dengan **DFS** atau **BFS**. Kenapa tidak melakukan *cheat* dengan indexing? Tambahkan sistem indexing untuk mempercepat search

Mau **B+ Tree?** Boleh, **Simple hash table?** Boleh, **DIY indexing system?** Why not?

Some fun fact: Meski B+ Tree biasanya hanya diajarkan pada mata kuliah **IF2240 - Basis Data** dan **IF3140 - Manajemen Basis Data**, *Actually* B+ Tree juga efisien untuk **File System**

- Kreativitas **(Up to 10)**

Karena setelah tahap shell sudah tidak beda jauh dengan *sandbox*. Implementasikan fitur yang tidak ada pada spesifikasi. Nilai akan bervariasi bergantung dengan assessment dari asisten

IV. Penilaian

Berikut adalah penilaian kelompok untuk milestone 3

- **Paging (20)**

- Struktur data Paging diimplementasikan (5)
- Kernel sudah di-link pada Higher Half (5)
- Paging bekerja dengan baik (10)

- **User Mode (20)**

- Inisier fungsional (5)
- Simple memory allocator (5)
- Berhasil menjalankan user program (10)

- **Shell (60)**

- Syscall berjalan (10)
- Behavior dasar shell sesuai dengan spesifikasi (10)
- Built-in utility commands (40; 5 per utility)

Seperti yang telah disebutkan pada milestone-milestone sebelumnya, **nilai kelompok** akan dilakukan **weighting berdasarkan peer dan penilaian asisten secara manual** sebelum menjadi **nilai individu**. Nilai individu akan ditambah nilai demo secara langsung

Oleh karena itu, **dimungkinkan untuk salah satu anggota mendapatkan nilai 100 dan anggota lain mendapatkan 0**, pada kasus-kasus ekstrim (menghilang, tidak dapat dikontak, tidak mengerjakan, tidak mau berkontribusi, dan sebagainya).

V. Pengumpulan dan Deliverables

1. Tugas ini akan melanjutkan milestone sebelumnya sehingga penggerjaan akan tetap menggunakan **GitHub Classroom yang sama dengan milestone 1 dan 2**.
2. Kreativitas dalam penggerjaan sangat dianjurkan untuk memperdalam pemahaman. Penilaian sepenuhnya didasarkan dari kriteria penilaian.
3. Lakukanlah commit yang wajar dan sesuai best practice (tidak semua kode satu commit).
4. Hindari menaruh binary file hasil kompilasi pada repository. Gunakan .gitignore untuk mengabaikan file-file tersebut.
5. Pengumpulan dilakukan dengan membuat **release** dengan tag **v3.0** pada repository yang telah kelompok Anda buat sebelum deadline. Jika terdapat revisi, tambahkan angka minor pada versi tag (**v3.1, v3.2, ..., v3.x**) Pastikan tag sesuai format. **Repository team yang tidak memiliki tag ini akan dianggap tidak mengumpulkan Milestone 3.**
6. Kami akan **menindaklanjuti segala bentuk kecurangan** yang terstruktur, masif, dan sistematis.
7. Hindari menggunakan fitur code auto-complete seperti Copilot, Codex, dll pada penggerjaan tugas besar ini sepenuhnya. Jika kode hasil tools tersebut diketahui tim asisten mengambil langsung dari internet, **praktikan yang menggunakan tools tersebut akan dianggap melakukan kecurangan** (Tanggung jawab kode akan berada pada praktikan, bukan tools auto-completion).
8. Jika ada pertanyaan atau masalah penggerjaan harap segera menggunakan sheets **QnA mata kuliah IF2230 Sistem Operasi** pada [link berikut](#).
9. Akan terdapat asistensi opsional bagi kelompok yang membutuhkan bantuan lebih lanjut. Setiap kelompok yang membutuhkan asistensi dapat mengisi form Asistensi IF2230 Sistem Operasi pada [form berikut](#). **Asisten akan mengontak maksimal paling lambat H+1 setelah request.**
10. Hindari menggunakan asistensi untuk **meminta bantuan mengimplementasikan kode**. Kerjakan terlebih dahulu sebisa mungkin. Gunakan asistensi untuk menanyakan konsep atau *bug* yang aneh
11. **Akan ada demo setelah penggerjaan milestone ke 3.** Informasi lebih lanjut untuk demo akan dirilis melalui milis

VI. Tips Pengerjaan

- Berbeda dengan milestone 2, guidebook dan kit milestone 3 akan dirilis secara bersamaan, berikut linknya

Dokumen Guidebook - [IF2230 - Guidebook - Milestone 3](#)

Kit - [Sister20/kit-OS-milestone-3-2023](#)

- Seperti yang disebutkan pada milestone sebelumnya, **milestone ini sangat dependen dengan kondisi milestone 1 dan milestone 2**. Pastikan milestone sebelumnya sudah berjalan dengan baik sebelum melanjutkan lebih jauh milestone ini.
- Pasang register **cs** dan **eip** pada watch expression debugger. Register **cs** dan **eip** akan menunjukkan posisi instruksi sekarang. Jika **cs** bernilai 0x8000 atau 0xF000, biasanya hal tersebut menandai kode default CPU exception handler.
- Jika milestone 2 File System berfokus kepada secondary storage, sebagian besar milestone ini akan berfokus ke memory. Gunakan debugger ketika merasa ada yang aneh dengan eksekusi program atau isi dari sebuah memory
- **Asistensi masih dibuka untuk milestone 3**
- **Pro tip:** Tugas besar ini ditujukan untuk pengerjaan secara kelompok dan jangka panjang.

Penilaian akhir akan menggunakan weighting, sehingga nilai individu akan berbeda-beda dalam satu kelompok. Weighting akan berdasarkan peer dan penilaian dari asisten case-by-case basis.

Jika tidak mengerjakan apapun, expect nilai individu kurang dari nilai kelompok, hingga untuk kasus ekstrim dapat di nol-kan untuk nilai individu

VII. Referensi

1. Kitab Intel x86 dan x64 - Intel® 64 and IA-32 Architectures Developer's Manual Vol 3a

<https://www.intel.com/content/www/us/en/architecture-and-technology/64-ia-32-architectures-software-developer-vol-3a-part-1-manual.html.html>

2. Guidebook IF2230 -  IF2230 - Guidebook - Milestone 3

3. <https://littleosbook.github.io>

4.  IF2230 - Debugger & WSL

“Mana pesan?”

~Dzaky~

Jekjekjekjekjek