

Spesifikasi Tugas Kecil
"The First Journey"
IF2230 - Sistem Operasi
Kompilasi Kernel Linux dari *Source* dan Implementasi *syscall*

Dipersiapkan oleh :
Asisten Lab Sistem Terdistribusi

Didukung Oleh :



Waktu Mulai :
Rabu, 6 Februari 2019, 18.00.00 WIB

Waktu Akhir :
Rabu, 13 Februari 2019, 12.59.59 WIB

I. Latar Belakang



Wahai para petualang! Kalian adalah para pemberani yang akan menjelajahi *The Abyss*! Namun, *The Abyss* sangatlah dalam, dan kalian membutuhkan peranti yang sesuai agar kalian dapat menjelajahnya.

Oleh karena itu, kalian membawa alat sonar berbasis linux untuk mengetahui kedalaman kalian saat ini di dalam *The Abyss*. Akan tetapi, beberapa *syscall* yang diperlukan ternyata rusak dalam perjalanan kalian, sehingga kalian perlu meng-*infuse* kernel linux kalian untuk memperbaikinya. Implementasikanlah *syscall* yang rusak ini dan temukanlah *the secret of The Abyss*!

Sedikit penjelasan, kernel dari sistem operasi merupakan bagian terpenting dari sistem operasi itu sendiri. Kernel adalah program yang pertama kali dijalankan ketika proses *booting* berlangsung. Kernel juga menjadi sebuah jembatan bagi perangkat keras untuk dapat berkomunikasi dengan sistem operasi. Salah satu kernel yang paling umum digunakan adalah kernel Linux ciptaan Linus Torvalds.

Sebuah *syscall* (singkatan dari *system call*) merupakan salah satu fungsi yang disediakan kernel Linux yang dapat dipanggil oleh program biasa. Gunanya yaitu agar program dapat mengakses layanan yang disediakan oleh kernel linux, seperti yang berhubungan dengan perangkat keras, penciptaan proses baru, dan lain-lain.

II. Deskripsi Tugas

1. Lakukan kompilasi kernel Linux dari source.
2. Lakukan booting ke kernel hasil kompilasi sebelumnya.
3. Tambahkan sebuah syscall baru pada source kernel Linux yang mencetak pesan "Hello, World!" ke log kernel.
 - *Prototype* fungsi *syscall* yang digunakan:

`asmlinkage long sys_hworld (void)`
 - **Bonus:** Selain mencetak pesan "Hello, World!", pada baris selanjutnya cobalah juga cetak pesan "Hello, World" tersebut dalam sebuah bahasa lain yang menggunakan huruf non-ASCII (Misal: "こんにちは、世界 !" dalam bahasa Jepang).
4. Pilihlah **salah satu** dari antara 3 fungsi di bawah ini. Implementasikan fungsi tersebut sebagai sebuah *syscall*.

- **Soal 1**

Prototype fungsi *syscall* yang digunakan:

`asmlinkage long sys_maxminup (int n, ...)`

Buatlah sebuah fungsi yang **mencari nilai maksimum, minimum, dan jumlah kenaikan** dari parameter kedua hingga parameter terakhir. Definisi dari jumlah kenaikan adalah banyaknya angka yang menaik (misal urutan 1,3,2,10,5,4,6,8,3,8 memiliki 4 kenaikan). Parameter n menentukan jumlah bilangannya. Hasil dicetak ke log kernel dengan urutan bilangan maksimum, lalu bilangan minimum, dan yang terakhir jumlah kenaikan.

- **Soal 2**

Prototype fungsi *syscall* yang digunakan:

`asmlinkage long sys_island (char* str)`

Buatlah sebuah fungsi yang **menentukan banyaknya "pulau"** yang direpresentasikan dalam string 1 dan 0. Pada string ini, bilangan 1 menandakan daratan dan bilangan 0 menandakan lautan. Sebagai

contoh, string "01100111001001111" memiliki 4 pulau. Cetak jumlah pulau ke dalam log kernel.

○ **Soal 3**

Prototype fungsi *syscall* yang digunakan:

```
asm linkage long sys_sort (int n, ...)
```

Buatlah sebuah fungsi yang **mengurutkan bilangan** dari parameter kedua hingga parameter terakhir dari fungsi tersebut. Jika parameter $n > 0$, maka pengurutan dilakukan dari bilangan terkecil ke bilangan terbesar (*ascending*) dan jumlah bilangan yang diurutkan sebanyak n . Jika parameter $n < 0$, maka pengurutan dilakukan dari bilangan terbesar ke bilangan terkecil (*descending*) dan jumlah bilangan yang diurutkan sebanyak $-n$. Hasil pengurutan dicetak per baris ke log kernel.

Perhatikan bahwa beberapa soal di sini memanfaatkan fungsi *variadic* yang menerima parameter berjumlah variabel. Referensi yang dapat digunakan : <http://en.cppreference.com/w/c/variadic>

III. Langkah Pengerjaan

A. Kompilasi kernel Linux dari source

1. Masuklah ke sistem operasi Linux anda. Anda boleh menggunakan Linux dual boot atau VM, tetapi kami anjurkan anda menggunakan VM supaya tidak terjadi kerusakan pada sistem anda. Siapkan storage sekitar 30GB untuk tugas ini.
2. Download archive tar.xz source Linux yang sama atau paling dekat dengan versi kernel Linux yang anda sekarang gunakan dari <https://www.kernel.org/pub/linux/kernel/>. Anda dapat menggunakan perintah `uname -r` pada terminal untuk mengetahui versi kernel anda.
3. Buka terminal anda di direktori downloads anda dan lakukan extract archive tersebut ke `/usr/src/` dengan perintah:
`sudo tar xf linux-<versi kernel>.tar.xz -C /usr/src/`
4. Pindah ke direktori `/usr/src/linux-<versi kernel>/` dengan perintah:
`cd /usr/src/linux-<versi kernel>/`

5. Lakukan konfigurasi Makefile source anda dengan perintah:
sudo make oldconfig

Screenshot hasil perintah tersebut untuk laporan anda.

6. Buka Makefile source anda dengan perintah **sudo nano Makefile**, dan ubah baris

```
EXTRAVERSION =
```

Menjadi

```
EXTRAVERSION = -K04-Kelompok01
```

Ganti **K04** dan **Kelompok01** menjadi kelas dan nomor kelompok anda.

7. Lakukan kompilasi source kernel Linux anda dengan perintah:
sudo make -j 4

Flag -j digunakan untuk melakukan parallel execution sehingga mempercepat proses kompilasi. Ganti angka 4 dengan jumlah core komputer/VM anda. Anda dapat menggunakan perintah nproc pada terminal untuk mengetahui jumlah core anda.

Perhatian: Gunakan GCC versi 4 jika terdapat masalah. Kompilasi kernel pertama kali dapat memakan waktu sekitar 1,5 jam dan menghasilkan direktori yang berukuran 8-14 gigabyte (tergantung hasil konfigurasi Makefile).

Screenshot hasil perintah tersebut untuk laporan anda.

8. Lakukan instalasi kernel anda dengan perintah:
sudo make modules_install install -j 4

Screenshot hasil perintah tersebut untuk laporan anda.

B. Booting ke kernel hasil kompilasi

1. Reboot komputer/VM anda.
2. Masuk ke grub (tekan tombol *shift* jika grub tidak muncul).
3. Pilih **Advanced Options**

4. Pilih **linux-<versi kernel>-K04-Kelompok1**.
Sesuaikan dengan versi dan extraversion kernel anda sebelumnya.
5. Buka terminal dan gunakan perintah **uname -r** untuk mengecek versi kernel anda.

Screenshot hasil perintah tersebut untuk laporan anda.

C. Penambahan *syscall* yang mencetak "Hello, World!" ke log kernel

1. Pindah ke direktori `/usr/src/linux-<versi kernel>/` dengan perintah:
cd /usr/src/linux-<versi kernel>/
2. Buatlah sebuah direktori baru bernama `hworld` dengan perintah:
sudo mkdir hworld
dan pindah ke direktori tersebut dengan perintah:
cd hworld
3. Buatlah sebuah file baru bernama `hworld.c` dengan perintah:
sudo nano hworld.c
yang berisi kode sebagai berikut:

```
#include <linux/kernel.h>

asmlinkage long sys_hworld (void) {
    printk("Hello, World!\n");
    return 0;
}
```

S.

4. Buatlah sebuah Makefile dengan perintah:
sudo nano Makefile
yang berisi kode berikut:

```
obj-y := hworld.o
```

5. Kembali ke root source kernel (dengan perintah **cd ..**) dan buka Makefile kernel dengan perintah:
sudo nano Makefile

Ubahlah baris

```
core-y += kernel/ mm/ fs/ ipc/ security/ crypto/ block/
```

menjadi

```
core-y += kernel/ mm/ fs/ ipc/ security/ crypto/ block/  
hworld/
```

Hal ini dilakukan supaya direktori hworld masuk ke dalam proses kompilasi kernel.

6. Buka *syscall table* dari kernel dengan perintah:
sudo nano arch/x86/entry/syscalls/syscall_32.tbl
Ganti dengan syscall_64.tbl jika komputer/VM anda 64-bit.

Tambahkan baris berikut di akhir file.

```
359 i386 hworld sys_hworld
```

Sesuaikan 359 sehingga merupakan angka sesudah dari baris sebelumnya (jika sebelumnya 358, maka 359) dan sesuaikan i386 sehingga sama dengan baris sebelumnya.

7. Buka file header syscall kernel dengan perintah:
sudo nano include/linux/syscalls.h

Tambahkan prototype fungsi di akhir file.

```
asmlinkage long sys_hworld (void);
```

8. Lakukan kompilasi kernel ulang dengan perintah:
sudo make -j 4 && sudo make modules_install install -j 4

Kompilasi kali ini akan lebih cepat dibanding kompilasi pertama kali (sekitar 10-15 menit).

Screenshot hasil perintah tersebut untuk laporan anda.

9. Reboot komputer/VM anda (jangan lupa untuk pilih kernel hasil kompilasi anda di grub).
10. Buatlah sebuah program C dengan nama test_hworld.c di sembarang direktori (jangan di root kernel) dengan kode sebagai berikut:

```
#include <stdio.h>
#include <linux/kernel.h>
#include <sys/syscall.h>

int main () {
    long int test = syscall(359);
    printf("sys_hworld output: %ld\n", test);
    return 0;
}
```

11. Lakukan kompilasi terhadap file tersebut.
12. Jalankan program hasil kompilasi. Jika implementasi *syscall* benar maka akan keluar output program seperti:
sys_hworld output: 0

Screenshot hasil output program tersebut untuk laporan anda.

13. Lihatlah isi log kernel di terminal dengan perintah:
dmesg

Jika implementasi *syscall* benar maka akan keluar output seperti:
[23164.192524] Hello, World!

Perhatian: Terkadang cetakan *printk* belum keluar saat menggunakan perintah *dmesg*. Coba jalankan program test lagi jika tidak terlihat pada *dmesg*.

14. **Bonus:** Ubah baris *printk* pada *hworld.c* sehingga pada baris cetakan selanjutnya juga terdapat "Hello, World!" dalam sebuah bahasa lain yang menggunakan huruf non-ASCII. Misal: "こんにちは、世界！" dalam bahasa Jepang.

Hint: perintah *dmesg* akan secara otomatis escape karakter-karakter non-ASCII sehingga output *dmesg* akan terlihat seperti berikut:
[23164.192524] Hello, World!

[23164.192524]

```
\xffffffffe3\xffffffff81\xffffffff93\xffffffff81\xffffffff93\xffffffe
3\xffffffff82\xffffffff93\xffffffff82\xffffffff93\xffffffe3\xffffff
81\xfffffffa\xffffffff81\xfffffffab\xffffffe3\xffffffff81\xffffff
a1\xffffffff81\xfffffffa1\xffffffe3\xffffffff81\xffffffaf\xfffff
f81\xffffffaf\xffffffe3\xffffff80\xffffffff81\xffffff80\xfffff
ff81\xffffffe4\xffffffb8\xffffff96\xffffffb8\xffffff96\xfffff
ffe7\xffffff95\xffffff8c\xffffff95\xffffff8c\xffffffef\xfff
ffffbc\xffffffff81\xffffffbc\xffffff81
```

Akan tetapi log kernel tidak hanya dapat dilihat dari perintah dmesg saja. Coba cari perintah/program/file lain sehingga dapat terlihat output log kernel yang tidak di-escape.

Screenshot hasil output dmesg dan juga hasil output perintah/program/file lain yang anda gunakan.

D. Penambahan *syscall* yang mengimplementasikan fungsi tertentu

Lakukan hal yang sama seperti pada bagian sebelumnya. Pilihlah hanya **salah satu** dari tiga fungsi yang diberikan. Anda tidak perlu mengerjakan semuanya.

Hint: Sebelum melakukan kompilasi kernel, coba implementasikan fungsi yang anda ingin buat di program C biasa terlebih dahulu sehingga dapat dipastikan fungsi anda berjalan dengan benar. Ganti fungsi printk dengan printf, dll.

IV. Pengumpulan dan *Deliverables*

- Buatlah sebuah archive zip dengan nama **IF2230-TugasKecil-K04-Kelompok1.zip**
Sesuaikan **K04** dan **Kelompok1** dengan kelas dan nomor kelompok Anda. Kumpulkan file zip tersebut di <http://bit.ly/PengumpulanTucilOS2019>
- Archive zip tersebut akan berisi
 - /src
 - /linux-<versi kernel>
 - Makefile
 - /arch/x86/syscalls
 - syscall_32.tbl atau syscall_64.tbl

- /include/linux
 - syscalls.h
 - /hworld
 - hworld.c
 - Makefile
 - /soal1 atau /soal2 atau /soal3
 - soal1.c atau soal2.c atau soal3.c
 - Makefile
 - /test
 - test_hworld.c
 - Test_soal1.c atau test_soal2.c atau test_soal3.c
 - /doc
 - IF2230-TugasKecil-K04-Kelompok1-Laporan.pdf
- Laporan dengan format PDF bernama IF2230-TugasKecil-K04-Kelompok1-Laporan.pdf akan berisi:
 - Cover
 - Deskripsi singkat permasalahan
 - Langkah pengerjaan (sertakan screenshot sesuai dengan yang diminta di spesifikasi).
 - Jawaban pertanyaan:
 - Mengapa lebih baik menggunakan source kernel Linux dengan versi yang sama atau dekat dengan kernel Linux anda sekarang?
 - Apa kegunaan dari perintah make, make modules_install, dan make install? Mengapa untuk kompilasi source linux tugas ini tidak digunakan perintah make modules?
 - Apa kegunaan dari Makefile syscall, syscall table, dan syscalls.h?