

Tugas Besar 1

IF2230 Sistem Operasi

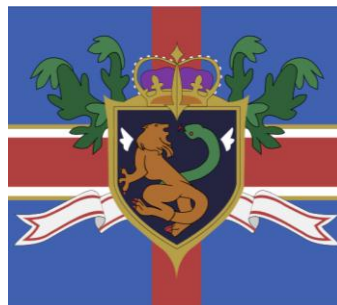
Implementasi System Call pada Kernel Linux 3.6++

Dipersiapkan oleh:

SISTER Assistants

Knightmare Frame R&D Team IT Division

Didukung oleh:



START: 17 Maret 2014

END: 1 April 2014

A. Story Background

Selamat datang, peserta magang tim R&D Nightmare Frame! Seperti yang anda ketahui belakangan ini makin sering terjadi serangan teroris, terutama setelah kemunculan Zero dan Order of Black Knight di Area 11. Oleh karena itu, untuk meningkatkan kualitas Nightmare Frame yang sudah ada, tim R&D membuat training dan magang untuk melatih orang-orang berbakat seperti anda bekerja pada industri ini.

Nightmare Frame, pasukan robot andalan Kekaisaran Britannia, memiliki beberapa komponen penting, salah satunya sistem operasi. Sistem operasi Nightmare Frame berbasis kernel linux dengan beberapa peningkatan untuk memenuhi standar militer Kekaisaran Britannia. Pada magang kali ini, anda diberikan tugas berkaitan dengan pengembangan fungsionalitas sistem operasi Nightmare Frame, yaitu implementasi system call pada kernel linux.

B. Deskripsi Tugas

Pada tugas ini, anda diminta untuk mengimplementasikan *system call* pada kernel linux 3.6 ke atas. Tugas ini dibuat dalam bahasa pemrograman C dan dengan compiler gcc. Tugas ini dapat dikerjakan pada lingkungan virtualisasi (misalnya VMWare atau VirtualBox), maupun langsung pada PC. Pada bagian akhir dokumen ini, terdapat panduan untuk membantu anda melakukan kompilasi kernel.

Adapun poin-poin tugas ini secara spesifik adalah sebagai berikut.

1. Buatlah sebuah system call untuk melakukan enkripsi dan dekripsi dengan *Caesarian chipper* dengan prototype sebagai berikut:

```
// fungsi enkripsi
asm linkage int sys_caesarenc(char *src, char *dest, char key);

// fungsi dekripsi
asm linkage int sys_caesardec(char *src, char *dest, char key);
```

dengan source adalah string sumber, dest adalah tempat menyimpan string hasil, dan key adalah besar pergeseran. Kedua fungsi tersebut mengembalikan 0.

Enkripsi dan dekripsi Caesarian chipper dilakukan dengan cara:

```
dest[i] = (src[i] + key) % 256; // enkripsi

dest[i] = (src[i] - key) % 256; // dekripsi
```

2. Buatlah sebuah test program yang berjalan pada user space yang memanggil kedua system call tersebut.
3. **(Bonus)** Buatlah sebuah system call untuk melakukan “komunikasi” primitif antar user proses. “Komunikasi” ini dilakukan dengan cara menyimpan sebuah int di memori (global variable) pada kernel space. Prototype system call tersebut adalah sebagai berikut.

```
// fungsi pemberian nilai
asm linkage int sys_setmyint(int value);

// fungsi pengambilan nilai
asm linkage int sys_getmyint(int *dest);
```

dengan value adalah int yang ingin disimpan, dan dest adalah tempat penerimaan int yg diambil. Setmyint mengembalikan:

- 1 jika data berhasil diubah (data yang lama sudah diambil dengan getmyint)
- 0 jika tidak diubah (data yang lama masih ada)

Getmyint mengembalikan:

- 1 jika isi dest berubah (ada data baru dari setmyint)
- 0 jika isi dest juga tidak diubah (tidak ada data baru)

Buatlah juga dua buah test program, program pertama melakukan “pengiriman” data (set integer), dan program kedua melakukan “penerimaan” data (get integer).

C. Pengumpulan Tugas

1. Daftarkan kelompok anda pada <http://beat.inkubatorit.com/os2014tb1>
2. Buatlah sebuah zip/rar dengan nama TB1_KX_YY, dengan KX nomor kelas (K1 atau K2) dan YY nomor kelompok sesuai daftar pada link di atas (00, 01, 02, dst). File zip/rar ini berisi
 - Folder src, berisi *source code* implementasi *system call* dan test program.
 - Folder bin, berisi *executable* test program dan hasil modifikasi kernel (file System.map, config, vmlinuz dan initrd.img pada /boot)
 - Folder doc, berisi laporan
3. Laporan berisi:
 - Cover
 - Deskripsi singkat permasalahan
 - Langkah pengerjaan, mulai dari kompilasi kernel hingga menambahkan kode system call ke kernel
 - Analisis, jelaskan bagian yang paling sulit / menarik saat pengerjaan beserta alasan dan penyelesaiannya (jika yang dijelaskan adalah yang sulit)
4. Tugas besar dikumpulkan pada **1 April 2014**. Prosedur pengumpulan akan diumumkan paling lambat H-1 deadline.

D. Panduan

Pada bagian ini akan dijelaskan cara kompilasi dan instalasi kernel linux 3.6 ke atas, khususnya kernel linux versi 3.13.6. Jika anda menggunakan kernel 3.6 ke atas lainnya, mungkin terdapat sedikit perbedaan. Panduan ini tidak ditulis dengan detail, sehingga jika terdapat kekurangan silahkan mencari tambahan literatur lain.

1. Unduh source code kernel 3.13.6 dari <https://www.kernel.org>

2. Ekstrak tar source code ke /usr/src

```
# tar -xJvf linux-3.13.6.tar.xz -C /usr/src
```
3. Pindah ke folder source code

```
# cd /usr/src/linux-3.13.6
```
4. Konfigurasi kode sebelum konfigurasi

untuk konfigurasi dengan setting saat ini:

```
# make oldconfig
```


untuk konfigurasi manual:

```
# make config
```
5. Kompilasi

```
# make
```
6. Instalasi

```
# make modules_install install
```
7. Update entri grub

```
# cd /boot  
# update-grub
```
8. Reboot
9. Saat masuk grub, pilih Advanced Options for Ubuntu (jika anda menggunakan distro Ubuntu), lalu pilih Ubuntu dengan kernel 3.13.6. Jika tidak berhasil masuk, boot ulang dan pilih Ubuntu kernel 3.13.6 dengan Recovery Mode.
10. Untuk mengecek apakah kernel yang anda gunakan adalah versi yang benar, gunakan

```
$ uname -r
```