

Laporan Tugas Besar 1

IF2230 Sistem Operasi

Disusun Oleh :

Victor Trimulya /13512038
Michael Alexander / 13512046
Kevin / 13512096



Program Studi Teknik Informatika
Institut Teknologi Bandung
Jl. Ganesha 10, Bandung 40132

A. Deskripsi Singkat

Pada tugas ini diminta untuk melakukan Implementasi system call pada kernel linux 3.6. 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
asmlinkage int sys_caesarenc(char *src, char *dest, char key);
```

```
// fungsi dekripsi
asmlinkage 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
asmlinkage int sys_setmyint(int value);
```

```
// fungsi pengambilan nilai
asmlinkage 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).

Langkah Pengerjaan

1. Download linux 3.13.7. dari <http://kernel.org>.
2. Ekstrak tar source code ke /usr/src
tar -xJvf linux-3.13.7.tar.xz -C /usr/src
3. Pindah ke folder source code
#cd /usr/src/linux-3 13.7
4. Konfigurasi kode sebelum konfigurasi untuk konfigurasi dengan setting saat ini:
make oldconfig
5. Edit /usr/src/linux-3 13.7/arch/x86/syscalls/syscall_64.tbl
6. Tambahkan sebelum "# x32-specific system call...".Dibawah entri terakhir pada kernel ini adalah 313 tambahkan:

```
314 64      caesarenc    sys_caesarenc
315 64      caesardec    sys_caesardec
316 64      setmyint     sys_setmyint
317 64      getmyint     sys_getmyint
```

7. Edit /usr/src/linux-3 13.7/include/linux/syscalls.h
Sebelum #endif tambahkan
asmlinkage int sys_caesarenc(char *src, char *dest, char key);
asmlinkage int sys_caesardec(char *src,char *dest, char key);
asmlinkage int sys_setmyint(int value);
asmlinkage int sys_getmyint(int *dest);
8. Ke Folder /usr/src/linux-3 13.7/kernel, Dibuat folder baru caesar dan comms
9. Di folder caesar dibuat 3 file Makefile,sys_caesardec.c,sys_caesarenc.c
10. Isi makefile dalam folder caesar :

```
obj-y := sys_caesarenc.o
obj-y += sys_caesardec.o
```

11. Dan untuk sys_caesardec.c diisi :
#include <linux/kernel.h>
#include <linux/syscalls.h>

```
asmlinkage int sys_caesardec(char *src,char *dest, char key) {
    int i;
    i = 0;
    while (src[i] != '\0') {
        dest[i] = (src[i] - key) % 256;
        i++;
    }
    return 0;
}
```

12. Dan untuk sys_caesarrenc.c diisi :
#include <linux/kernel.h>
#include <linux/syscalls.h>

```
asmlinkage int sys_caesarenc(char *src, char *dest, char key){
    int i;
    i = 0;
    while (src[i] != '\0') {
        dest[i] = (src[i] + key) % 256;
        i++;
    }
    return 0;
}
```

- ```

 }
13. Masuk ke dalam folder comms(untuk soal bonus), buatlah makefile dengan isi
 obj-y := sys_comms.o
 dan sys_comms.c dengan isi :
 #include <linux/kernel.h>
 #include <linux/syscalls.h>

 int CharlesZiBritannia = -9999;

 asmlinkage int sys_setmyint(int value){
 if (CharlesZiBritannia == -9999) {
 CharlesZiBritannia = value;
 return 1;
 } else {
 return 0;
 }
 }

 asmlinkage int sys_getmyint(int *dest){
 if (CharlesZiBritannia != -9999) {
 *dest = CharlesZiBritannia;
 CharlesZiBritannia = -9999;
 return 1;
 } else {
 return 0;
 }
 }
 }
14. Edit Makefile yang ada di dalam folder kernel
 Setelah line terakhir obj-y += ... tambahkan
 obj-y += caesar/
 obj-y += comms/
15. Kembali ke folder source code
 # cd /usr/src/linux-3.13.7
16. Kompilasi dengan
 # make
17. Setelah selesai kompilasi
 # make modules_install install
18. Update entri grub
 # cd /boot
 # update-grub
19. Reboot
20. Lakukan pembuatan driver untuk testing system call.

```

### **Bagian yang paling menarik/sulit**

Kompilasi kernel selalu berhasil pada pertama kali percobaan dengan mengikuti tutorial yang ada di <http://xathrya.web.id/blog/2012/10/16/add-new-system-call-to-linux-3-6-1/> dengan penyesuaian untuk system 64 bit sehingga kami tidak mengalami kesulitan yang berarti kecuali lama waktu kompilasi sekitar 2 jam untuk setiap kali compile. Kami melakukan 3 kali kompilasi, yang pertama kompilasi kernel fresh tanpa modifikasi, lalu kompilasi kedua untuk implementasi soal pertama dan kompilasi ketiga untuk kompilasi soal bonus.