Universidad del Valle de Guatemala
Sistemas Operativos
Jose Block 18935

Laboratorio#1

Ejercicio#1

La variación del número es debido a que no es en el mismo número de proceso en el que se hace, por lo que el proceso cambia de ID. Se puede notar que se le va agregando 1 a cada proceso, debido a que se pidió uno detrás del otro.

Por ser un fork(), se estan ejecutando dos procesos a la vez por lo que hay dos PIDs y en uno de estos se esta llamando el pid del programa anterior por lo que se repite un PID.

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1	root	20	0	2036	708	628	S	0.0	0.2	0:01.04	init
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	migration/0
4	root	20	0	0	0	0	S	0.0	0.0	0:00.03	ksoftirqd/0
5	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	watchdog/0
6	root	20	0	0	0	0	S	0.0	0.0	0:00.34	events/0
7	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuset
8	root	20	0	0	0	0	S	0.0	0.0	0:00.00	khelper
9	root	20	0	0	0	0	S	0.0	0.0	0:00.00	netns
10	root	20	0	0	0	0	S	0.0	0.0	0:00.00	async/mgr
11	root	20	0	0	0	0	S	0.0	0.0	0:00.00	pm
12	root	20	0	0	0	0	S	0.0	0.0	0:00.03	sync_supers
13	root	20	0	0	0	0	S	0.0	0.0	0:00.04	bdi-default
14	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kintegrityd/0
15	root	20	0	0	0	0	S	0.0	0.0	0:00.03	kblockd/0
16	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kacpid
17	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kacpi_notify

El primer proceso es init, es el proceso padre principal. Este proceso se mantiene corriendo hasta que el sistema se apague, y es iniciado por el kernel al encender el sistema.

Ejercicio#2

a.
- open():
  Abre el archivo especificado por nombre de ruta.
- close():
  Cierra un archivo, de modo que ya no se refiere a cualquier archivo y se puede reutilizar.
- read():
  ```
  ssize_t read(int fd, void *buf, size_t count);
  ```
  Intenta leer hasta count bytes del archivo fd en el buffer comenzando en buf.
- write():

  ```
  ssize_t write(int fd, const void *buf, size_t count);
  ```

  Escribe hasta count bytes desde el búfer comenzando en buf al archivo al que hace referencia el descriptor de archivo fd.

b.

```c
#include<stdio.h>
#include<unistd.h>

int main(int argc, char* argv[]){
        char linea[1024];
        FILE *readfile;
        FILE *writefile;

        readfile = fopen(argv[1], "r");
        writefile = fopen(argv[2], "w");
        while(fgets(linea,1024, (FILE*) readfile)){
                fprintf(writefile, "%s", linea);
        }
        fclose(readfile);
        fclose(writefile);
}
```

c.

```
os@debian:~$ sudo apt-get install strace
[sudo] password for os:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  strace
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 122 kB of archives.
After this operation, 303 kB of additional disk space will be used.
WARNING: The following packages cannot be authenticated!
  strace
Install these packages without verification [y/N]?
```

```
Install these packages without verification [y/N]? y
Err http://ftp.us.debian.org/debian/ squeeze/main strace i386 4.5.20-2
  404  Not Found [IP: 208.80.154.15 80]
Failed to fetch http://ftp.us.debian.org/debian/pool/main/s/strace/strace_4.5.20
-2_i386.deb  404  Not Found [IP: 208.80.154.15 80]
E: Unable to fetch some archives, maybe run apt-get update or try with --fix-mis
sing?
```

Después de las modificaciones:

```
os@debian:/etc/apt$ sudo apt-get install strace
Reading package lists... Done
Building dependency tree
Reading state information... Done
strace is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 228 not upgraded.
os@debian:/etc/apt$
```

d.

```
os@debian:~$ gcc -o ejercicio2 Ejercicio2.c
os@debian:~$ strace ./ejercicio2 original Copia_1
execve("./ejercicio2", ["./ejercicio2", "original", "Copia_1"], [/* 34 vars */])
 = 0
brk(0)                                  = 0x9c09000
access("/etc/ld.so.nohwcap", F_OK)      = -1 ENOENT (No such file or directory)
mmap2(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0xb7
770000
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or directory)
open("/etc/ld.so.cache", O_RDONLY)      = 3
fstat64(3, {st_mode=S_IFREG|0644, st_size=64828, ...}) = 0
mmap2(NULL, 64828, PROT_READ, MAP_PRIVATE, 3, 0) = 0xb7760000
close(3)                                = 0
access("/etc/ld.so.nohwcap", F_OK)      = -1 ENOENT (No such file or directory)
open("/lib/i686/cmov/libc.so.6", O_RDONLY) = 3
read(3, "\177ELF\1\1\1\0\0\0\0\0\0\0\0\0\3\0\3\0\1\0\0\0\0n\1\0004\0\0\0"..., 51
2) = 512
fstat64(3, {st_mode=S_IFREG|0755, st_size=1327556, ...}) = 0
mmap2(NULL, 1337704, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0xb
7619000
mprotect(0xb7759000, 4096, PROT_NONE)   = 0
mmap2(0xb775a000, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRI
TE, 3, 0x140) = 0xb775a000
mmap2(0xb775d000, 10600, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMO
US, -1, 0) = 0xb775d000
close(3)                                = 0
mmap2(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0xb7
618000
set_thread_area({entry_number:-1 -> 6, base_addr:0xb76186c0, limit:1048575, seg_
32bit:1, contents:0, read_exec_only:0, limit_in_pages:1, seg_not_present:0, usea
ble:1}) = 0
mprotect(0xb775a000, 8192, PROT_READ)   = 0
mprotect(0xb778e000, 4096, PROT_READ)   = 0
munmap(0xb7760000, 64828)               = 0
brk(0)                                  = 0x9c09000
brk(0x9c2a000)                          = 0x9c2a000
open("original", O_RDONLY)              = 3
open("Copia_1", O_WRONLY|O_CREAT|O_TRUNC, 0666) = 4
fstat64(3, {st_mode=S_IFREG|0644, st_size=10, ...}) = 0
mmap2(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0xb7
76f000
read(3, "Holakeace\n", 4096)            = 10
fstat64(4, {st_mode=S_IFREG|0644, st_size=0, ...}) = 0
mmap2(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0xb7
76e000
read(3, "", 4096)                       = 0
close(3)                                = 0
munmap(0xb776f000, 4096)                = 0
write(4, "Holakeace\n", 10)             = 10
close(4)                                = 0
munmap(0xb776e000, 4096)                = 0
```

Ejercicio#3

a.
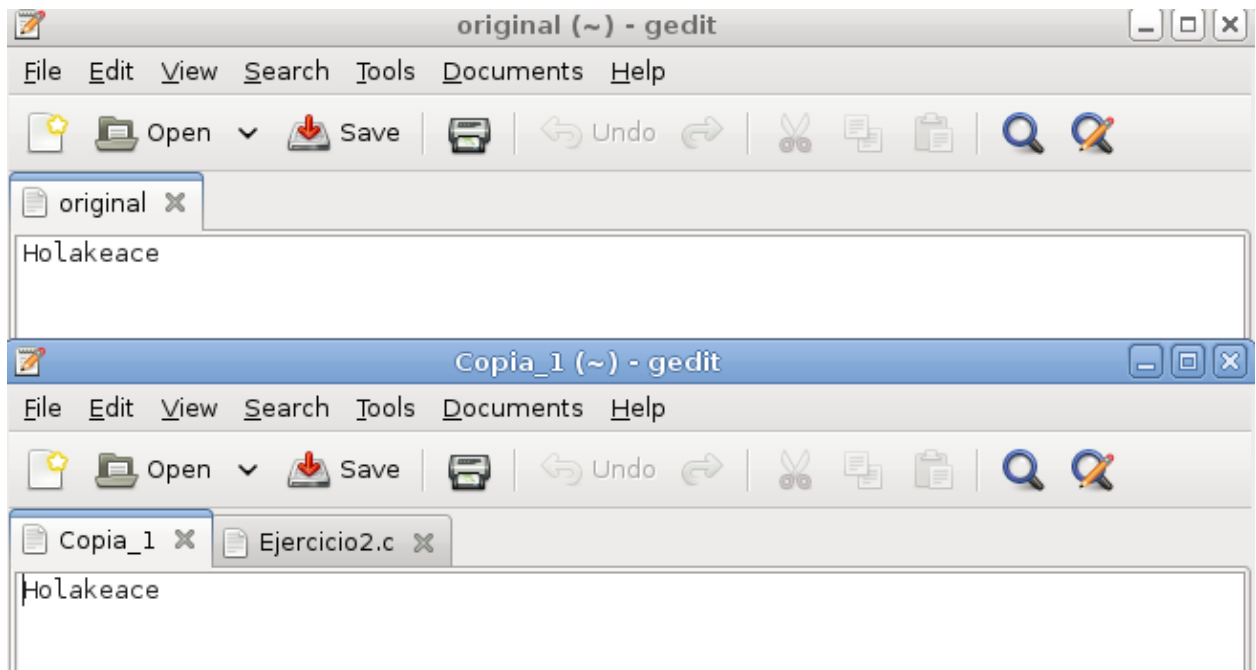
```
ps@debian:~$ cd ~
ps@debian:~$ sudo cp -a /usr/src/linux-2.6.39.4 .
ps@debian:~$
```

b.

```
File  Edit  View  Terminal  Help
  GNU nano 2.2.4                    File: syscall_table_32.S

        .long sys_fallocate
        .long sys_timerfd_settime        /* 325 */
        .long sys_timerfd_gettime
        .long sys_signalfd4
        .long sys_eventfd2
        .long sys_epoll_create1
        .long sys_dup3                   /* 330 */
        .long sys_pipe2
        .long sys_inotify_init1
        .long sys_preadv
        .long sys_pwritev
        .long sys_rt_tgsigqueueinfo      /* 335 */
        .long sys_perf_event_open
        .long sys_recvmmsg
        .long sys_fanotify_init
        .long sys_fanotify_mark
        .long sys_prlimit64              /* 340 */
        .long sys_name_to_handle_at
        .long sys_open_by_handle_at
        .long sys_clock_adjtime
        .long sys_syncfs
        .long sys_mycall                 /*345*/
```
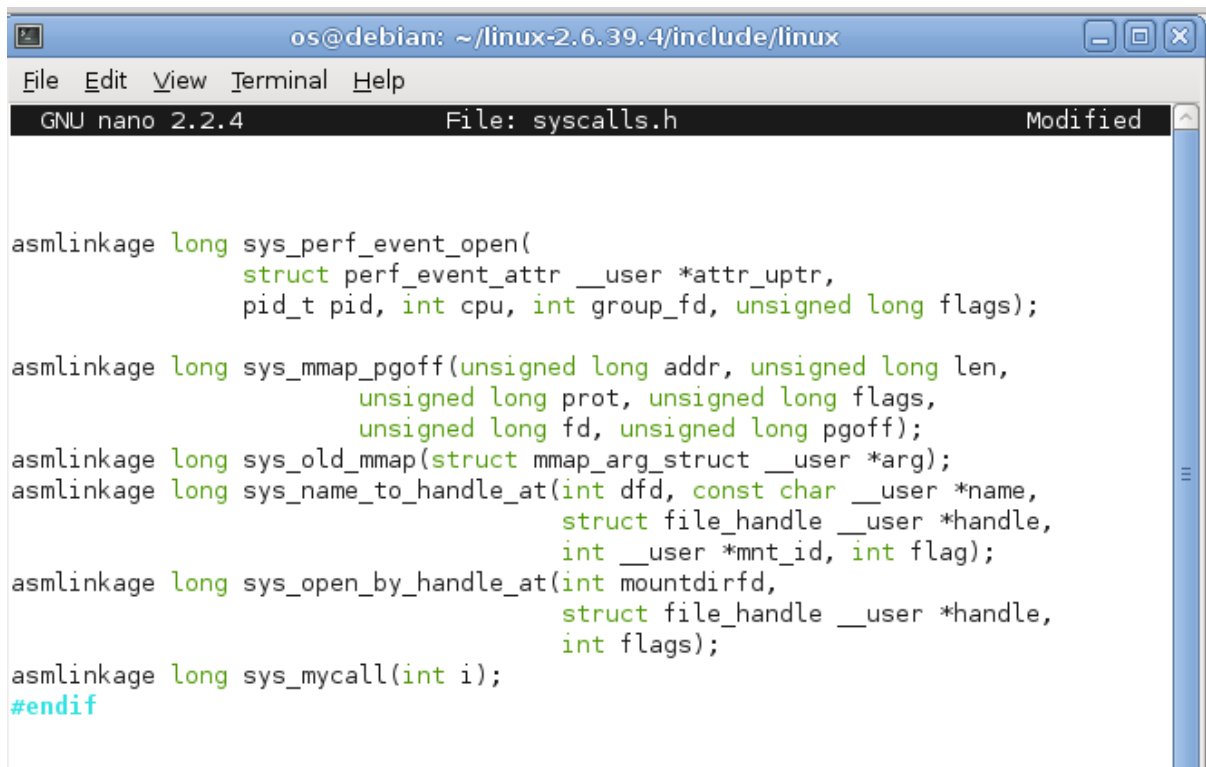
c.

```
#define __NR_pipe2                  331
#define __NR_inotify_init1          332
#define __NR_preadv                 333
#define __NR_pwritev                334
#define __NR_rt_tgsigqueueinfo      335
#define __NR_perf_event_open        336
#define __NR_recvmmsg               337
#define __NR_fanotify_init          338
#define __NR_fanotify_mark          339
#define __NR_prlimit64              340
#define __NR_name_to_handle_at      341
#define __NR_open_by_handle_at      342
#define __NR_clock_adjtime          343
#define __NR_syncfs                 344
#define __NR_mycall                 345

#ifdef __KERNEL__

#define NR_syscalls 346

#define __ARCH_WANT_IPC_PARSE_VERSION
#define __ARCH_WANT_OLD_READDIR
#define __ARCH_WANT_OLD_STAT
#define __ARCH_WANT_STAT64
```

d.

```
os@debian: ~/linux-2.6.39.4/include/linux

File   Edit   View   Terminal   Help
  GNU nano 2.2.4              File: syscalls.h                    Modified


asmlinkage long sys_perf_event_open(
                struct perf_event_attr __user *attr_uptr,
                pid_t pid, int cpu, int group_fd, unsigned long flags);

asmlinkage long sys_mmap_pgoff(unsigned long addr, unsigned long len,
                    unsigned long prot, unsigned long flags,
                    unsigned long fd, unsigned long pgoff);
asmlinkage long sys_old_mmap(struct mmap_arg_struct __user *arg);
asmlinkage long sys_name_to_handle_at(int dfd, const char __user *name,
                                      struct file_handle __user *handle,
                                      int __user *mnt_id, int flag);
asmlinkage long sys_open_by_handle_at(int mountdirfd,
                                      struct file_handle __user *handle,
                                      int flags);
asmlinkage long sys_mycall(int i);
#endif
```

e.

File Edit View Terminal Help

```
  GNU nano 2.2.4              File: myca

#include<linux/linkage.h>
asmlinkage long sys_mycall(int i){
        return i+13;
}
```

File Edit View Terminal Help

```
  GNU nano 2.2.4              File: Makefile

obj-y := mycall.o
```

f. F

g. F

h. f