REPORT - ASSIGNMENT 2

Q1:annapooranihello

Q2:annapooraniprint

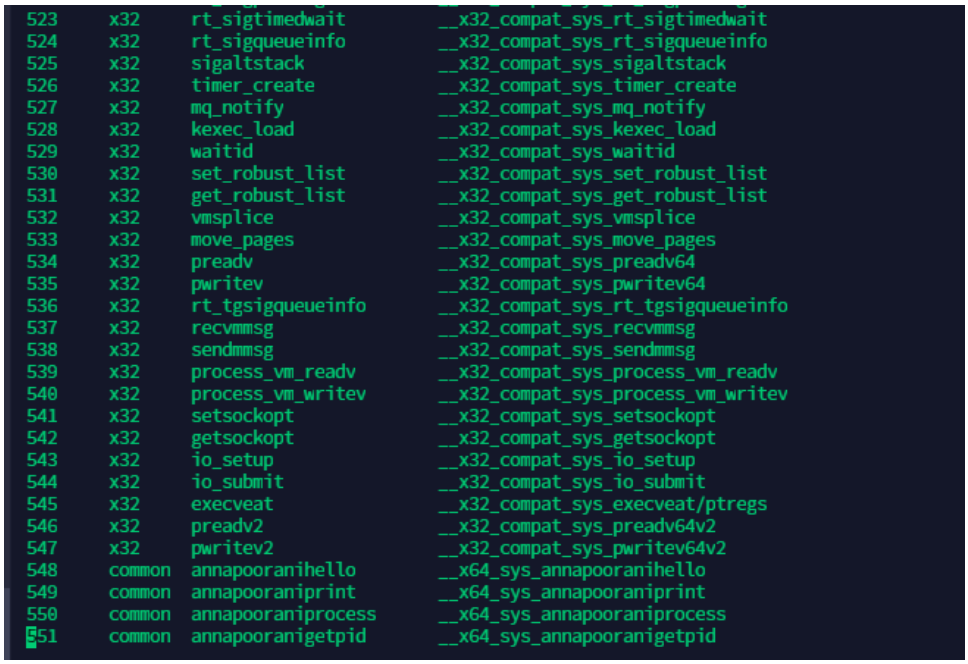
Q3:annapooraniprocess

Q4:annapooranigetpid

STEPS:

After installation:

1. Go to " cd arch/x86/entry/syscalls/" and open " vi syscall_64.tbl file". Add the four system calls and notice the number. Mine is from 548 to 551.



```
523     x32     rt_sigtimedwait         __x32_compat_sys_rt_sigtimedwait
524     x32     rt_sigqueueinfo         __x32_compat_sys_rt_sigqueueinfo
525     x32     sigaltstack             __x32_compat_sys_sigaltstack
526     x32     timer_create            __x32_compat_sys_timer_create
527     x32     mq_notify               __x32_compat_sys_mq_notify
528     x32     kexec_load              __x32_compat_sys_kexec_load
529     x32     waitid                  __x32_compat_sys_waitid
530     x32     set_robust_list         __x32_compat_sys_set_robust_list
531     x32     get_robust_list         __x32_compat_sys_get_robust_list
532     x32     vmsplice                __x32_compat_sys_vmsplice
533     x32     move_pages              __x32_compat_sys_move_pages
534     x32     preadv                  __x32_compat_sys_preadv64
535     x32     pwritev                 __x32_compat_sys_pwritev64
536     x32     rt_tgsigqueueinfo       __x32_compat_sys_rt_tgsigqueueinfo
537     x32     recvmmsg                __x32_compat_sys_recvmmsg
538     x32     sendmmsg                __x32_compat_sys_sendmmsg
539     x32     process_vm_readv        __x32_compat_sys_process_vm_readv
540     x32     process_vm_writev       __x32_compat_sys_process_vm_writev
541     x32     setsockopt              __x32_compat_sys_setsockopt
542     x32     getsockopt              __x32_compat_sys_getsockopt
543     x32     io_setup                __x32_compat_sys_io_setup
544     x32     io_submit               __x32_compat_sys_io_submit
545     x32     execveat                __x32_compat_sys_execveat/ptregs
546     x32     preadv2                 __x32_compat_sys_preadv64v2
547     x32     pwritev2                __x32_compat_sys_pwritev64v2
548     common  annapooranihello        __x64_sys_annapooranihello
549     common  annapooraniprint        __x64_sys_annapooraniprint
550     common  annapooraniprocess      __x64_sys_annapooraniprocess
551     common  annapooranigetpid       __x64_sys_annapooranigetpid
```

2. From the linux-4.19.210/ folder, go to "/include/linux/" and open "vi syscalls.h"

Add the following lines at the bottom for the 4 system calls for the 4 questions.

```
 * should or should not be restarted, but returns the raw error codes from
 * __close_fd().
 */
static inline int ksys_close(unsigned int fd)
{
        return __close_fd(current->files, fd);
}

extern long do_sys_open(int dfd, const char __user *filename, int flags,
                        umode_t mode);

static inline long ksys_open(const char __user *filename, int flags,
                             umode_t mode)
{
        if (force_o_largefile())
                flags |= O_LARGEFILE;
        return do_sys_open(AT_FDCWD, filename, flags, mode);
}

extern long do_sys_truncate(const char __user *pathname, loff_t length);

static inline long ksys_truncate(const char __user *pathname, loff_t length)
{
        return do_sys_truncate(pathname, length);
}

static inline unsigned int ksys_personality(unsigned int personality)
{
        unsigned int old = current->personality;

        if (personality != 0xffffffff)
                set_personality(personality);

        return old;
}
asmlinkage int sys_annapooranihello(void);
asmlinkage int sys_annapooraniprint(char*);
asmlinkage int sys_annapooraniprocess(void);
asmlinkage int sys_annapooranigetpid(void);


#endif
```

3: Go to kernel directory. There, create the following files:

Q1: create annapooranihello.c [path: linux-4.19.210/kernel/annapooranihello.c]

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

SYSCALL_DEFINE0(annapooranihello)
{
    printk("Hello!\n");
    return 0;
}

~
~
~
~
~
```

Q2:create annapooraniprint.c

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

SYSCALL_DEFINE1(annapooraniprint, char*,s)
{

    char buf[256];
    long msg= strncpy_from_user(buf, s, sizeof(buf));
    if( msg< 0 || msg== sizeof(buf))
        return -EFAULT;
    printk("Annapoorani printing string:  %s\n", buf);
    return 0;

}
```

Q3:create annapooraniprocess.c

```c
#include <linux/syscalls.h>
#include <linux/kernel.h>
#include<linux/sched.h>
#include<linux/cred.h>
SYSCALL_DEFINE0(annapooraniprocess)
{
    printk("Parent pid: %d",current->parent->pid);
    printk("Current pid: %d",current->pid);
    return 0;
}
```

Q4:create annapooranigetpid.c

```c
#include <linux/syscalls.h>
#include <linux/kernel.h>
#include<linux/sched.h>
#include<linux/cred.h>

SYSCALL_DEFINE0(annapooranigetpid)
{
    return task_tgid_vnr(current);
}
```

4. Add the following in the Makefile in the kernel directory [path: linux-4.19.210/kernel/Makefile]

```
# SPDX-License-Identifier: GPL-2.0
#
# Makefile for the linux kernel.
#

obj-y     = fork.o exec_domain.o panic.o \
            cpu.o exit.o softirq.o resource.o \
            sysctl.o sysctl_binary.o capability.o ptrace.o user.o \
            signal.o sys.o umh.o workqueue.o pid.o task_work.o \
            extable.o params.o \
            kthread.o sys_ni.o nsproxy.o \
            notifier.o ksysfs.o cred.o reboot.o \
            async.o range.o smpboot.o ucount.o annapooranihello.o annapooraniprint.o annapooraniprocess.o annapooranigetpid.o

obj-$(CONFIG_MODULES) += kmod.o
obj-$(CONFIG_MULTIUSER) += groups.o
```

annapooranihello.o annapooraniprint.o annapooraniprocess.o annapooranigetpid.o

5. Run the following command from the linux-4.19.210/ directory]:

cp -v /boot/config-$(uname -r) .config

6. Open config file and set CONFIG_SYSTEM_TRUSTED_KEYS as empty string

```
#
# Certificates for signature checking
#
CONFIG_MODULE_SIG_KEY="certs/signing_key.pem"
CONFIG_SYSTEM_TRUSTED_KEYRING=y
CONFIG_SYSTEM_TRUSTED_KEYS=""
CONFIG_SYSTEM_EXTRA_CERTIFICATE=y
CONFIG_SYSTEM_EXTRA_CERTIFICATE_SIZE=4096
CONFIG_SECONDARY_TRUSTED_KEYRING=y
CONFIG_SYSTEM_BLACKLIST_KEYRING=y
CONFIG_SYSTEM_BLACKLIST_HASH_LIST=""
CONFIG_BINARY_PRINTF=y

#
# Library routines
#
```

6. Run sudo make olddefconfig

7. Run the folowing commands:

sudo make prepare

sudo make

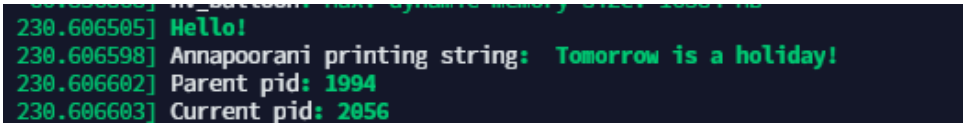
sudo make modules_install

sudo make install

8. Finaly perform sudo reboot

9.Run a c file calling the syscalls.

NOTE: for question 3, the process ids are different for current process and parent process. This shows the processes are not the same.



for the fourth question, the pid value is returned.