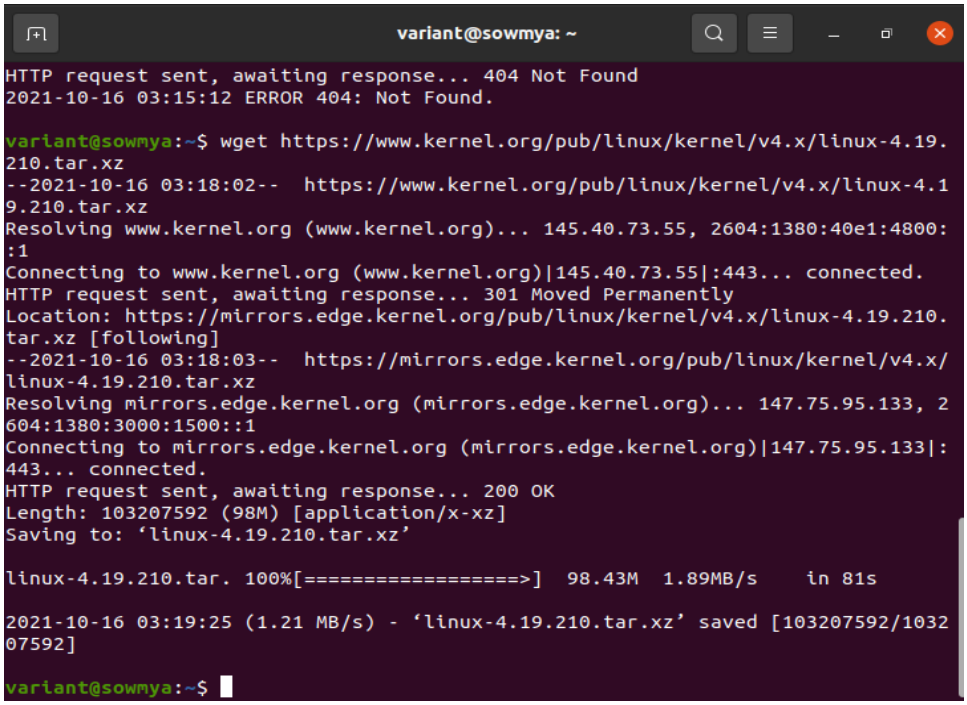
AOS Assignment 2

Adding and testing a new system call to Linux kernel

Common Steps:

1. Download the kernel with version 4.19.210.



```
variant@sowmya: ~
HTTP request sent, awaiting response... 404 Not Found
2021-10-16 03:15:12 ERROR 404: Not Found.

variant@sowmya:~$ wget https://www.kernel.org/pub/linux/kernel/v4.x/linux-4.19.
210.tar.xz
--2021-10-16 03:18:02--  https://www.kernel.org/pub/linux/kernel/v4.x/linux-4.1
9.210.tar.xz
Resolving www.kernel.org (www.kernel.org)... 145.40.73.55, 2604:1380:40e1:4800:
:1
Connecting to www.kernel.org (www.kernel.org)|145.40.73.55|:443... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://mirrors.edge.kernel.org/pub/linux/kernel/v4.x/linux-4.19.210.
tar.xz [following]
--2021-10-16 03:18:03--  https://mirrors.edge.kernel.org/pub/linux/kernel/v4.x/
linux-4.19.210.tar.xz
Resolving mirrors.edge.kernel.org (mirrors.edge.kernel.org)... 147.75.95.133, 2
604:1380:3000:1500::1
Connecting to mirrors.edge.kernel.org (mirrors.edge.kernel.org)|147.75.95.133|:
443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 103207592 (98M) [application/x-xz]
Saving to: ‘linux-4.19.210.tar.xz’

linux-4.19.210.tar. 100%[===================>]  98.43M  1.89MB/s    in 81s

2021-10-16 03:19:25 (1.21 MB/s) - ‘linux-4.19.210.tar.xz’ saved [103207592/1032
07592]

variant@sowmya:~$
```

2. Extract the content of the downloaded tar file to the directory /usr/src

```
variant@sowmya: ~
linux-4.19.210/virt/kvm/arm/vgic/trace.h
linux-4.19.210/virt/kvm/arm/vgic/vgic-debug.c
linux-4.19.210/virt/kvm/arm/vgic/vgic-init.c
linux-4.19.210/virt/kvm/arm/vgic/vgic-irqfd.c
linux-4.19.210/virt/kvm/arm/vgic/vgic-its.c
linux-4.19.210/virt/kvm/arm/vgic/vgic-kvm-device.c
linux-4.19.210/virt/kvm/arm/vgic/vgic-mmio-v2.c
linux-4.19.210/virt/kvm/arm/vgic/vgic-mmio-v3.c
linux-4.19.210/virt/kvm/arm/vgic/vgic-mmio.c
linux-4.19.210/virt/kvm/arm/vgic/vgic-mmio.h
linux-4.19.210/virt/kvm/arm/vgic/vgic-v2.c
linux-4.19.210/virt/kvm/arm/vgic/vgic-v3.c
linux-4.19.210/virt/kvm/arm/vgic/vgic-v4.c
linux-4.19.210/virt/kvm/arm/vgic/vgic.c
linux-4.19.210/virt/kvm/arm/vgic/vgic.h
linux-4.19.210/virt/kvm/async_pf.c
linux-4.19.210/virt/kvm/async_pf.h
linux-4.19.210/virt/kvm/coalesced_mmio.c
linux-4.19.210/virt/kvm/coalesced_mmio.h
linux-4.19.210/virt/kvm/eventfd.c
linux-4.19.210/virt/kvm/irqchip.c
linux-4.19.210/virt/kvm/kvm_main.c
linux-4.19.210/virt/kvm/vfio.c
linux-4.19.210/virt/kvm/vfio.h
linux-4.19.210/virt/lib/
linux-4.19.210/virt/lib/Kconfig
linux-4.19.210/virt/lib/Makefile
linux-4.19.210/virt/lib/irqbypass.c
variant@sowmya:~$
```

3. Go to the /usr/src/linux-4.19.210 directory

```
variant@sowmya: /usr/src/linux-4.19.210
variant@sowmya:~$ cd /usr/src
variant@sowmya:/usr/src$ ls
linux-4.19.210                    linux-hwe-5.11-headers-5.11.0-37
linux-headers-5.11.0-37-generic   linux-hwe-5.8-headers-5.8.0-43
linux-headers-5.8.0-43-generic
variant@sowmya:/usr/src$ cd linux-4.19.210
variant@sowmya:/usr/src/linux-4.19.210$
```

4. Do required changes and run make command to compile the kernel – sudo make
5. After compiling the kernel, install the kernel – sudo make modules_install install

```
variant@sowmya:/usr/src/linux-4.19.210$ sudo make -j3
[sudo] password for variant:
  DESCEND  objtool
  CALL    scripts/checksyscalls.sh
  CHK     include/generated/compile.h
  Building modules, stage 2.
Kernel: arch/x86/boot/bzImage is ready  (#13)
  MODPOST 4990 modules
variant@sowmya:/usr/src/linux-4.19.210$ sudo make modules_install install
[sudo] password for variant:
Sorry, try again.
[sudo] password for variant:
  INSTALL arch/x86/crypto/aegis128-aesni.ko
  INSTALL arch/x86/crypto/aes-x86_64.ko
  INSTALL arch/x86/crypto/aesni-intel.ko
```

6. Restart and while restarting press the shift button to get grub menu. Select the required kernel from the grub menu.

Ques 1: Write syscall to print welcome message to Linux logs

1.Create the directory sowmyahello in /usr/src/linux-4.19.210 folder

```
variant@sowmya: /usr/src/linux-4.19.210/sowmyahello
variant@sowmya:/usr/src/linux-4.19.210$ sudo mkdir sowmyahello
variant@sowmya:/usr/src/linux-4.19.210$ cd sowmyahello
variant@sowmya:/usr/src/linux-4.19.210/sowmyahello$ ls
variant@sowmya:/usr/src/linux-4.19.210/sowmyahello$ vi sowmyahello.c
```

2. Create sowmyahello.c file in the directory and add the below code.

```
variant@sowmya: /usr/src/linux-4.19.210/sowmyahello
#include<linux/kernel.h>
asmlinkage int sys_sowmyahello(void)
{
        printk("Hello Sowmya!!!");
        return 0;
}
~
```

3. Create a Makefile in the same directory and add the below line

```
variant@sowmya: /usr/src/linux-4.19.210/sowmyahello
obj-y := sowmyahello.o
~
~
~
~
~
```

4. In the Makefile of the kernel folder add the name of the system call you are creating

```
variant@sowmya: /usr/src/linux-4.19.210
variant@sowmya: /usr/src/linux-4.19.210    variant@sowmya: /usr/src/linux-4.19.210
variant@sowmya:/usr/src/linux-4.19.210$ pwd
/usr/src/linux-4.19.210
variant@sowmya:/usr/src/linux-4.19.210$ sudo vi Makefile
[sudo] password for variant:
variant@sowmya:/usr/src/linux-4.19.210$
```

```
  else
    SKIP_STACK_VALIDATION := 1
    export SKIP_STACK_VALIDATION
  endif
endif

PHONY += prepare0

ifeq ($(KBUILD_EXTMOD),)
core-y          += kernel/ certs/ mm/ fs/ ipc/ security/ crypto/ block/ hello/ sowmyahello/
```

5. In the syscall_64.tbl file in the below specified path, add the entry of new system call.

```
variant@sowmya:/usr/src/linux-4.19.210$ pwd
/usr/src/linux-4.19.210
variant@sowmya:/usr/src/linux-4.19.210$ sudo vi Makefile
[sudo] password for variant:
variant@sowmya:/usr/src/linux-4.19.210$ cd arch/x86/entry/syscalls
variant@sowmya:/usr/src/linux-4.19.210/arch/x86/entry/syscalls$ ls
Makefile  syscall_32.tbl  syscall_64.tbl  syscallhdr.sh  syscalltbl.sh
variant@sowmya:/usr/src/linux-4.19.210/arch/x86/entry/syscalls$ sudo gedit syscall_64.tbl
```

```
379 538     x32     sendmmsg                __x32_compat_sys_sendmmsg
380 539     x32     process_vm_readv        __x32_compat_sys_process_vm_readv
381 540     x32     process_vm_writev       __x32_compat_sys_process_vm_writev
382 541     x32     setsockopt              __x32_compat_sys_setsockopt
383 542     x32     getsockopt              __x32_compat_sys_getsockopt
384 543     x32     io_setup                __x32_compat_sys_io_setup
385 544     x32     io_submit               __x32_compat_sys_io_submit
386 545     x32     execveat                __x32_compat_sys_execveat/ptregs
387 546     x32     preadv2                 __x32_compat_sys_preadv64v2
388 547     x32     pwritev2                __x32_compat_sys_pwritev64v2
389 548     64      hello                   sys_hello
390 549     64      sowmyahello             sys_sowmyahello
```

7.In the syscalls.h header file, add the signature of the system call

```
variant@sowmya: /usr/src/linux-4.19.210/include/linux
variant@sowmya: /usr/src/linux-4.19.210/includ...   variant@sowmya: /usr/src/linux-4.19.210/includ...
variant@sowmya:/usr/src/linux-4.19.210/arch/x86/entry/syscalls$ cd ../../../..
variant@sowmya:/usr/src/linux-4.19.210$ pwd
/usr/src/linux-4.19.210
variant@sowmya:/usr/src/linux-4.19.210$ cd include/linux
variant@sowmya:/usr/src/linux-4.19.210/include/linux$ sudo gedit syscalls.h
```

8.After doing above steps, compile the kernel.

```
variant@sowmya:/usr/src$ cd linux-4.19.210/
variant@sowmya:/usr/src/linux-4.19.210$ ls
arch        CREDITS        fs       Kbuild    MAINTAINERS      Module.symvers  security      sowmyaprocess  vmlinux
block       crypto         hello    Kconfig   Makefile         net             sound         System.map     vmlinux-gdb.py
built-in.a  Documentation  include  kernel    mm               README          sowmyagetpid  tools          vmlinux.o
certs       drivers        init     lib       modules.builtin  samples         sowmyahello   usr
COPYING     firmware       ipc      LICENSES  modules.order    scripts         sowmyaprint   virt
variant@sowmya:/usr/src/linux-4.19.210$ sudo make -j3
[sudo] password for variant:
  DESCEND  objtool
  CALL    scripts/checksyscalls.sh
```

9.Install the changes made to the kernel

```
variant@sowmya:/usr/src/linux-4.19.210$ sudo make -j3
[sudo] password for variant:
  DESCEND  objtool
  CALL    scripts/checksyscalls.sh
  CHK     include/generated/compile.h
  Building modules, stage 2.
Kernel: arch/x86/boot/bzImage is ready  (#13)
  MODPOST 4990 modules
variant@sowmya:/usr/src/linux-4.19.210$ sudo make modules_install install
[sudo] password for variant:
Sorry, try again.
[sudo] password for variant:
  INSTALL arch/x86/crypto/aegis128-aesni.ko
  INSTALL arch/x86/crypto/aes-x86_64.ko
  INSTALL arch/x86/crypto/aesni-intel.ko
```

10. Restart the system

11. Test the system call. The number assigned to first system call is 549.

```
  GNU nano 4.8                                                    shello.c
#include <sys/syscall.h>
#include <stdio.h>
#include <linux/kernel.h>
#include <unistd.h>
int main()
{
        long x=syscall(549);
        printf("The value returned by system call hello is %ld",x);
        return 0;
}
```

12. The result of the system call can be seen in kernel logs using dmesg command

```
[ 3845.293370] ohci-pci 0000:00:06.0: frame counter not updating; disabled
[ 3845.293381] ohci-pci 0000:00:06.0: HC died; cleaning up
[ 3845.293864] usb 1-1: USB disconnect, device number 2
[ 8559.136819] Message printed by sowmyahello system call: Hello Sowmya!!!
variant@sowmya:~/aos$
```

Ques 2: Write syscall which will receive string parameter and print it along with some message to kernel logs.

1.Create the directory sowmyaprint in /usr/src/linux-4.19.210 folder

```
variant@sowmya:/usr/src/linux-4.19.210$ ls
arch        CREDITS        fs       Kbuild     MAINTAINERS      Module.symvers  security      sowmyaprocess  vmlinux
block       crypto         hello    Kconfig    Makefile         net             sound         System.map     vmlinux-gdb.py
built-in.a  Documentation  include  kernel     mm               README          sowmyagetpid  tools          vmlinux.o
certs       drivers        init     lib        modules.builtin  samples         sowmyahello   usr
COPYING     firmware       ipc      LICENSES   modules.order    scripts         sowmyaprint   virt
variant@sowmya:/usr/src/linux-4.19.210$
```

2. Create sowmyaprint.c file in the directory and add the below code.

```
  GNU nano 4.8                                          sowmyaprint.c
#include <linux/kernel.h>
#include <linux/uaccess.h>
#include <linux/slab.h>
#include <linux/sched.h>
#include <asm/uaccess.h>
#include <linux/syscalls.h>

SYSCALL_DEFINE1(sowmyaprint,char *,string)
{
        char kstring[256];
        long x=strncpy_from_user(kstring,string,256);
        printk("The string passed to sys_sowmyaprint system call is %s\n",kstring);
        //char kstring2[512];
        //int y=copy_from_user(kstring2,(char *)string,512);
        //printk("The string passed to sys_sowmyaprint system call is %s\n",kstring2);
        return 0;
}
```

3. Create a Makefile in the same directory and add the below line

```
  GNU nano 4.8                                          Makefile
obj-y := sowmyaprint.o
```

4. In the Makefile of the kernel folder add the name of the system call you are creating

Makefile [Read-Only]
/usr/src/linux-4.19.210

```
ifeq ($(has_libelf),1)
  objtool_target := tools/objtool FORCE
else
  SKIP_STACK_VALIDATION := 1
  export SKIP_STACK_VALIDATION
endif
endif

PHONY += prepare0

ifeq ($(KBUILD_EXTMOD),)
core-y          += kernel/ certs/ mm/ fs/ ipc/ security/ crypto/ block/ hello/ sowmyahello/ sowmyaprint/ sowmyaprocess/ sowmyagetpid/

vmlinux-dirs    := $(patsubst %/,%,$(filter %/, $(init-y) $(init-m) \
                     $(core-y) $(core-m) $(drivers-y) $(drivers-m) \
                     $(net-y) $(net-m) $(libs-y) $(libs-m) $(virt-y)))
```

5. In the syscall_64.tbl file in the below specified path, add the entry of new system call.

syscall_64.tbl [Read-Only]
/usr/src/linux-4.19.210/arch/x86/entry/syscalls

```
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     64      hello                   sys_hello
549     64      sowmyahello             sys_sowmyahello
550     64      sowmyaprint             __x64_sys_sowmyaprint
551     64      sowmyaprocess           sys_sowmyaprocess
552     64      sowmyagetpid            sys_sowmyagetpid
```

Plain Text Tab Width: 8 Ln 1, Col 1 INS

7.In the syscalls.h header file, add the signature of the system call

```
variant@sowmya: /usr/src/linux-4.19.210/include/linux
variant@sowmya: /usr/src/linux-4.19.210/includ...   variant@sowmya: /usr/src/linux-4.19.210/includ...
variant@sowmya:/usr/src/linux-4.19.210/arch/x86/entry/syscalls$ cd ../../../..
variant@sowmya:/usr/src/linux-4.19.210$ pwd
/usr/src/linux-4.19.210
variant@sowmya:/usr/src/linux-4.19.210$ cd include/linux
variant@sowmya:/usr/src/linux-4.19.210/include/linux$ sudo gedit syscalls.h
```

```
Open   *syscalls.h   /usr/src/linux-4.19.210/include/linux   Save
1288        unsigned int old = current->personality;
1289
1290        if (personality != 0xffffffff)
1291                set_personality(personality);
1292
1293        return old;
1294 }
1295 asmlinkage long sys_hello(void);
1296 asmlinkage long sys_sowmyahello(void);
1297 asmlinkage long sys_sowmyaprint(const char __user *string);
1298 asmlinkage long sys_sowmyaprocess(void);
C/ObjC Header   Tab Width: 8   Ln 1298, Col 41   INS
```

8.After doing above steps, compile the kernel.

```
variant@sowmya:/usr/src$ cd linux-4.19.210/
variant@sowmya:/usr/src/linux-4.19.210$ ls
arch        CREDITS        fs       Kbuild    MAINTAINERS      Module.symvers  security      sowmyaprocess  vmlinux
block       crypto         hello    Kconfig   Makefile         net             sound         System.map     vmlinux-gdb.py
built-in.a  Documentation  include  kernel    mm               README          sowmyagetpid  tools          vmlinux.o
certs       drivers        init     lib       modules.builtin  samples         sowmyahello   usr
COPYING     firmware       ipc      LICENSES  modules.order    scripts         sowmyaprint   virt
variant@sowmya:/usr/src/linux-4.19.210$ sudo make -j3
[sudo] password for variant:
  DESCEND  objtool
  CALL    scripts/checksyscalls.sh
```

9.Install the changes made to the kernel

```
variant@sowmya:/usr/src/linux-4.19.210$ sudo make -j3
[sudo] password for variant:
  DESCEND  objtool
  CALL    scripts/checksyscalls.sh
  CHK     include/generated/compile.h
  Building modules, stage 2.
Kernel: arch/x86/boot/bzImage is ready  (#13)
  MODPOST 4990 modules
variant@sowmya:/usr/src/linux-4.19.210$ sudo make modules_install install
[sudo] password for variant:
Sorry, try again.
[sudo] password for variant:
  INSTALL arch/x86/crypto/aegis128-aesni.ko
  INSTALL arch/x86/crypto/aes-x86_64.ko
  INSTALL arch/x86/crypto/aesni-intel.ko
```

10. Restart the system

11. Test the system call. The number assigned to first system call is 550.

```
GNU nano 4.8                                                    strhello.c
#include <sys/syscall.h>
#include <stdio.h>
#include <linux/kernel.h>
#include <unistd.h>
int main()
{
        const char str[19]="Second_system_call";
        long x=syscall(550,&str);
        printf("The value returned by system call hello is %ld\n",x);
        return 0;
}
```

12. The result of the system call can be seen in kernel logs using dmesg command

```
[ 8586.398590] Message printed by sowmyahello system call: Hello Sowmya!!!
[ 9240.017246] The string passed to sys_sowmyaprint system call is Second_system_call
variant@sowmya:~/aos$
```

Ques 3: Write system call to print the parent process id and current process id upon calling it

1.Create the directory sowmyaprocess in /usr/src/linux-4.19.210 folder

```
variant@sowmya:/usr/src/linux-4.19.210$ ls
arch        CREDITS        fs       Kbuild    MAINTAINERS      Module.symvers  security      sowmyaprocess  vmlinux
block       crypto         hello    Kconfig   Makefile         net             sound         System.map     vmlinux-gdb.py
built-in.a  Documentation  include  kernel    mm               README          sowmyagetpid  tools          vmlinux.o
certs       drivers        init     lib       modules.builtin  samples         sowmyahello   usr
COPYING     firmware       ipc      LICENSES  modules.order    scripts         sowmyaprint   virt
variant@sowmya:/usr/src/linux-4.19.210$
```

2. Create sowmyaprocess.c file in the directory and add the below code.

```
GNU nano 4.8                                          sowmyaprocess.c
#include <linux/kernel.h>
#include <linux/cred.h>
#include <linux/sched.h>
#include <linux/syscalls.h>

asmlinkage long sys_sowmyaprocess(void)
{
        struct task_struct *cur=current;
        printk("The process id of the current process is %d\n",cur->pid);
        printk("The process id of the parent process of the current process is %d\n",cur->parent->pid);
        printk("The process id of the real parent of the current process id %d\n",cur->real_parent->pid);
        return 0;
}
```

3. Create a Makefile in the same directory and add the below line

```
GNU nano 4.8                                                    Makefile
obj-y := sowmyaprocess.o
```

4. In the Makefile of the kernel folder add the name of the system call you are creating

Makefile [Read-Only]
/usr/src/linux-4.19.210

```
ifeq ($(has_libelf),1)
  objtool_target := tools/objtool FORCE
else
  SKIP_STACK_VALIDATION := 1
  export SKIP_STACK_VALIDATION
endif
endif

PHONY += prepare0

ifeq ($(KBUILD_EXTMOD),)
core-y          += kernel/ certs/ mm/ fs/ ipc/ security/ crypto/ block/ hello/ sowmyahello/ sowmyaprint/ sowmyaprocess/ sowmyagetpid/

vmlinux-dirs    := $(patsubst %/,%,$(filter %/, $(init-y) $(init-m) \
                     $(core-y) $(core-m) $(drivers-y) $(drivers-m) \
                     $(net-y) $(net-m) $(libs-y) $(libs-m) $(virt-y)))
```

5. In the syscall_64.tbl file in the below specified path, add the entry of new system call.

syscall_64.tbl [Read-Only]
/usr/src/linux-4.19.210/arch/x86/entry/syscalls

```
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     64      hello                   sys_hello
549     64      sowmyahello             sys_sowmyahello
550     64      sowmyaprint             __x64_sys_sowmyaprint
551     64      sowmyaprocess           sys_sowmyaprocess
552     64      sowmyagetpid            sys_sowmyagetpid
```

Plain Text Tab Width: 8 Ln 1, Col 1 INS

7.In the syscalls.h header file, add the signature of the system call

```
variant@sowmya: /usr/src/linux-4.19.210/include/linux
variant@sowmya: /usr/src/linux-4.19.210/includ...   variant@sowmya: /usr/src/linux-4.19.210/includ...
variant@sowmya:/usr/src/linux-4.19.210/arch/x86/entry/syscalls$ cd ../../../..
variant@sowmya:/usr/src/linux-4.19.210$ pwd
/usr/src/linux-4.19.210
variant@sowmya:/usr/src/linux-4.19.210$ cd include/linux
variant@sowmya:/usr/src/linux-4.19.210/include/linux$ sudo gedit syscalls.h
```

```
Open   *syscalls.h   /usr/src/linux-4.19.210/include/linux   Save
1288        unsigned int old = current->personality;
1289
1290        if (personality != 0xffffffff)
1291                set_personality(personality);
1292
1293        return old;
1294 }
1295 asmlinkage long sys_hello(void);
1296 asmlinkage long sys_sowmyahello(void);
1297 asmlinkage long sys_sowmyaprint(const char __user *string);
1298 asmlinkage long sys_sowmyaprocess(void);
C/ObjC Header   Tab Width: 8   Ln 1298, Col 41   INS
```

8.After doing above steps, compile the kernel.

```
variant@sowmya:/usr/src$ cd linux-4.19.210/
variant@sowmya:/usr/src/linux-4.19.210$ ls
arch        CREDITS        fs       Kbuild    MAINTAINERS      Module.symvers  security      sowmyaprocess  vmlinux
block       crypto         hello    Kconfig   Makefile         net             sound         System.map     vmlinux-gdb.py
built-in.a  Documentation  include  kernel    mm               README          sowmyagetpid  tools          vmlinux.o
certs       drivers        init     lib       modules.builtin  samples         sowmyahello   usr
COPYING     firmware       ipc      LICENSES  modules.order    scripts         sowmyaprint   virt
variant@sowmya:/usr/src/linux-4.19.210$ sudo make -j3
[sudo] password for variant:
  DESCEND  objtool
  CALL    scripts/checksyscalls.sh
```

9.Install the changes made to the kernel

```
variant@sowmya:/usr/src/linux-4.19.210$ sudo make -j3
[sudo] password for variant:
  DESCEND  objtool
  CALL    scripts/checksyscalls.sh
  CHK     include/generated/compile.h
  Building modules, stage 2.
Kernel: arch/x86/boot/bzImage is ready  (#13)
  MODPOST 4990 modules
variant@sowmya:/usr/src/linux-4.19.210$ sudo make modules_install install
[sudo] password for variant:
Sorry, try again.
[sudo] password for variant:
  INSTALL arch/x86/crypto/aegis128-aesni.ko
  INSTALL arch/x86/crypto/aes-x86_64.ko
  INSTALL arch/x86/crypto/aesni-intel.ko
```

10. Restart the system

11. Test the system call. The number assigned to first system call is 550.

```
  GNU nano 4.8                                                    process.c
#include <sys/syscall.h>
#include <stdio.h>
#include <linux/kernel.h>
#include <unistd.h>
int main()
{
        long x=syscall(551);
        printf("The value returned by system call hello is %ld\n",x);
        return 0;
}
```

12. The result of the system call can be seen in kernel logs using dmesg command

```
[ 9240.017246] The string passed to sys_sowmyaprint system call is Second_system_call
[ 9507.425015] The process id of the current process is 15408
[ 9507.425018] The process id of the parent process of the current process is 1915
[ 9507.425019] The process id of the real parent of the current process id 1915
variant@sowmya:~/aos$
```

Observations: The process id's printed are different and below are my observations

1. In the above screenshot showing the results, the process id 15408 is the id the process which is invoking the sowmyaprocess system call.
2. The process id 1915 is the id of the process which invoked the current process ie..bash shell in this case.
3. Additional: When the parent process is killed/terminated before the completion of child process, the child process becomes an orphan. In that case another parent will be assigned (which can be init process).

Ques 4: Write system call to execute some predefined system call from your written system call

1.Create the directory sowmyagetpid in /usr/src/linux-4.19.210 folder

```
variant@sowmya:/usr/src/linux-4.19.210$ ls
arch        CREDITS        fs       Kbuild    MAINTAINERS      Module.symvers  security      sowmyaprocess  vmlinux
block       crypto         hello    Kconfig   Makefile         net             sound         System.map     vmlinux-gdb.py
built-in.a  Documentation  include  kernel    mm               README          sowmyagetpid  tools          vmlinux.o
certs       drivers        init     lib       modules.builtin  samples         sowmyahello   usr
COPYING     firmware       ipc      LICENSES  modules.order    scripts         sowmyaprint   virt
variant@sowmya:/usr/src/linux-4.19.210$
```

2. Create sowmyagetpid.c file in the directory and add the below code.

```
  GNU nano 4.8                                                    sowmyagetpid.c
#include <linux/kernel.h>
#include <linux/sched.h>
#include <linux/syscalls.h>
asmlinkage long sys_sowmyagetpid(void)
{
        int pid=task_tgid_vnr(current);
        printk("The result of sowmyagetpid system call is %d\n",pid);
        return (long)pid;
}
```

3. Create a Makefile in the same directory and add the below line

```
  GNU nano 4.8
obj-y := sowmyagetpid.o
```

4. In the Makefile of the kernel folder add the name of the system call you are creating

```
Open                Makefile [Read-Only]                Save
                    /usr/src/linux-4.19.210
ifeq ($(has_libelf),1)
  objtool_target := tools/objtool FORCE
else
  SKIP_STACK_VALIDATION := 1
  export SKIP_STACK_VALIDATION
endif
endif

PHONY += prepare0

ifeq ($(KBUILD_EXTMOD),)
core-y          += kernel/ certs/ mm/ fs/ ipc/ security/ crypto/ block/ hello/ sowmyahello/ sowmyaprint/ sowmyaprocess/
sowmyagetpid/

vmlinux-dirs    := $(patsubst %/,%,$(filter %/, $(init-y) $(init-m) \
                     $(core-y) $(core-m) $(drivers-y) $(drivers-m) \
                     $(net-y) $(net-m) $(libs-y) $(libs-m) $(virt-y)))
```

5. In the syscall_64.tbl file in the below specified path, add the entry of new system call.

```
Open                syscall_64.tbl [Read-Only]                Save
                    /usr/src/linux-4.19.210/arch/x86/entry/syscalls
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     64      hello                   sys_hello
549     64      sowmyahello             sys_sowmyahello
550     64      sowmyaprint             __x64_sys_sowmyaprint
551     64      sowmyaprocess           sys_sowmyaprocess
552     64      sowmyagetpid            sys_sowmyagetpid
                                Plain Text   Tab Width: 8   Ln 1, Col 1   INS
```

7.In the syscalls.h header file, add the signature of the system call

```
variant@sowmya: /usr/src/linux-4.19.210/include/linux
variant@sowmya: /usr/src/linux-4.19.210/includ...   variant@sowmya: /usr/src/linux-4.19.210/includ...
variant@sowmya:/usr/src/linux-4.19.210/arch/x86/entry/syscalls$ cd ../../../..
variant@sowmya:/usr/src/linux-4.19.210$ pwd
/usr/src/linux-4.19.210
variant@sowmya:/usr/src/linux-4.19.210$ cd include/linux
variant@sowmya:/usr/src/linux-4.19.210/include/linux$ sudo gedit syscalls.h
```

```
Open   syscalls.h   /usr/src/linux-4.19.210/include/linux   Save
        if (personality != 0xffffffff)
                set_personality(personality);

        return old;
}
asmlinkage long sys_hello(void);
asmlinkage long sys_sowmyahello(void);
asmlinkage long sys_sowmyaprint(const char __user *string);
asmlinkage long sys_sowmyaprocess(void);
asmlinkage long sys_sowmyagetpid(void);
#endif
```

8.After doing above steps, compile the kernel.

```
variant@sowmya:/usr/src$ cd linux-4.19.210/
variant@sowmya:/usr/src/linux-4.19.210$ ls
arch        CREDITS        fs       Kbuild    MAINTAINERS      Module.symvers  security      sowmyaprocess  vmlinux
block       crypto         hello    Kconfig   Makefile         net             sound         System.map     vmlinux-gdb.py
built-in.a  Documentation  include  kernel    mm               README          sowmyagetpid  tools          vmlinux.o
certs       drivers        init     lib       modules.builtin  samples         sowmyahello   usr
COPYING     firmware       ipc      LICENSES  modules.order    scripts         sowmyaprint   virt
variant@sowmya:/usr/src/linux-4.19.210$ sudo make -j3
[sudo] password for variant:
  DESCEND  objtool
  CALL    scripts/checksyscalls.sh
```

9.Install the changes made to the kernel

```
variant@sowmya:/usr/src/linux-4.19.210$ sudo make -j3
[sudo] password for variant:
  DESCEND  objtool
  CALL    scripts/checksyscalls.sh
  CHK     include/generated/compile.h
  Building modules, stage 2.
Kernel: arch/x86/boot/bzImage is ready  (#13)
  MODPOST 4990 modules
variant@sowmya:/usr/src/linux-4.19.210$ sudo make modules_install install
[sudo] password for variant:
Sorry, try again.
[sudo] password for variant:
  INSTALL arch/x86/crypto/aegis128-aesni.ko
  INSTALL arch/x86/crypto/aes-x86_64.ko
  INSTALL arch/x86/crypto/aesni-intel.ko
```

10. Restart the system

11. Test the system call. The number assigned to first system call is 550.

```
  GNU nano 4.8                                                    getpid.c
#include <sys/syscall.h>
#include <stdio.h>
#include <linux/kernel.h>
#include <unistd.h>
int main()
{
        long x=syscall(552);
        printf("The value returned by system call hello is %ld\n",x);
        return 0;
}
```

12. The result of the system call can be seen in kernel logs using dmesg command

```
variant@sowmya:~/aos$ nano getpid.c
variant@sowmya:~/aos$ cc getpid.c
variant@sowmya:~/aos$ ./a.out
The value returned by system call hello is 15433
variant@sowmya:~/aos$
```

```
[ 9507.425019] The process id of the real parent of the current process id 1915
[ 9911.882334] The result of sowmyagetpid system call is 15433
variant@sowmya:~/aos$
```

Note for the 4th question:

System calls are an interface between userspace and kernel space and should not be used from the kernel space. So, instead of using getpid system call, task_tgid_vnr function of the kernel library is used to get the process id of the current process. The system call getpid itself used the same function.