**ADDING SYSTEM CALLS**

1.Download the kernel 4.19.210

2.Extract the contents

3.Open the system call table at arch/x86/entry/syscalls/syscall\_64.tbl

and add the system call entries

(548,549,550,551)

A screenshot of a computer

Description automatically generated with medium confidence

4.Open the system call header file at include/linux/syscalls.h and add the prototypes of the functions. The asmlinkage says that arguments to the functions are from the stack

For example:

asmlinkage int sys\_my\_syscall\_0(void);

asmlinkage int sys\_my\_syscall\_1(int);

asmlinkage int sys\_my\_syscall\_2(int, int);

A screenshot of a computer

Description automatically generated with medium confidence

5.Now define the system calls in kernel space or any other directory

SYSCALL\_DEFINE#(functionname, datatype,argument,…)

{

//body

}

System call to print a message

A screenshot of a computer

Description automatically generated with medium confidence

System call that takes a string parameter and prints it

A screenshot of a computer

Description automatically generated with medium confidence

System call that prints current and parent process ids

A screenshot of a computer

Description automatically generated with medium confidence

System call that executes predefined system call

A screenshot of a computer

Description automatically generated with medium confidence

6.Change in Makefile

If the .c files are directly in kernel folder add the object file names to obj-y

If the files are in some other directory, add a makefile to the directory

7.Compile the kernel

sudo make prepare

sudo make -j5

8.Install the modules

sudo make -j5 modules\_install

sudo make install

9.Reboot the system(VM in this case)

sudo reboot

10.Perform the system calls

A screenshot of a computer

Description automatically generated with medium confidence

11.use $dmesg to view the kernel logs

A screenshot of a computer

Description automatically generated with medium confidence

For Q3

Both the process id’s are different. Current pid is the process id for the current running program in execution. Parent process is the bash process which runs the program.

Use $ps command for the parent process id

azureuser@AOS-system-calls:~/linux-4.19.210$ ps 5104

PID TTY STAT TIME COMMAND

5104 pts/0 Ss 0:00 -bash