COMP30640: Operating Systems – Finbar Allan

Assignment 4

Question 1

A copy of the xv6-riscv folder was created and named xv6-copy to implement a new system calling for printing Student ID: 22208108. The kernel syscall files were modified to add the `sys_hello()` system call handler function and this function was registered in the syscall array. Figure 1 shows a screenshot of the new system call handler function from the virtual machine; a copy of all files which have been updated to achieve the required output are included in this submission.

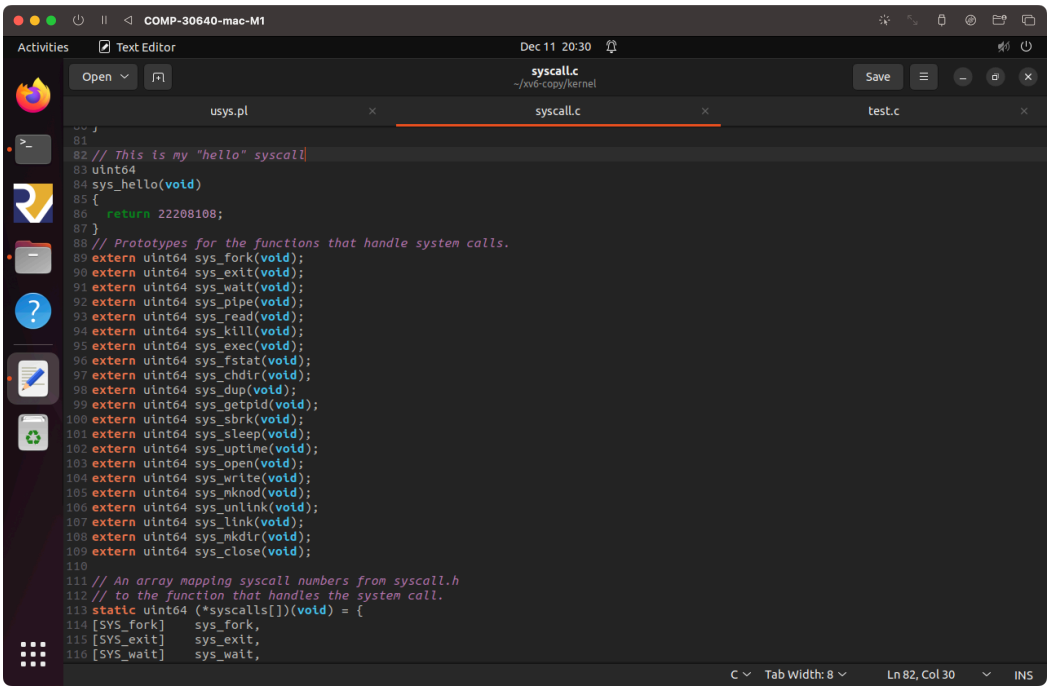

Figure 1: syscall.c file updated to include `sys_hello()` function.

To make the above system call handler function accessible the usys.py file must be modified to generate a user-mode C function which is achieved by including the line highlighted below. Figure 2 displays a screenshot of this line in the Perl file from the virtual environment.

```
entry("hello")
```

Similarly, the Makefile must be modified to include the following entry for programs which are accessible to the user (UPROGS):

```
$U/_test\
```

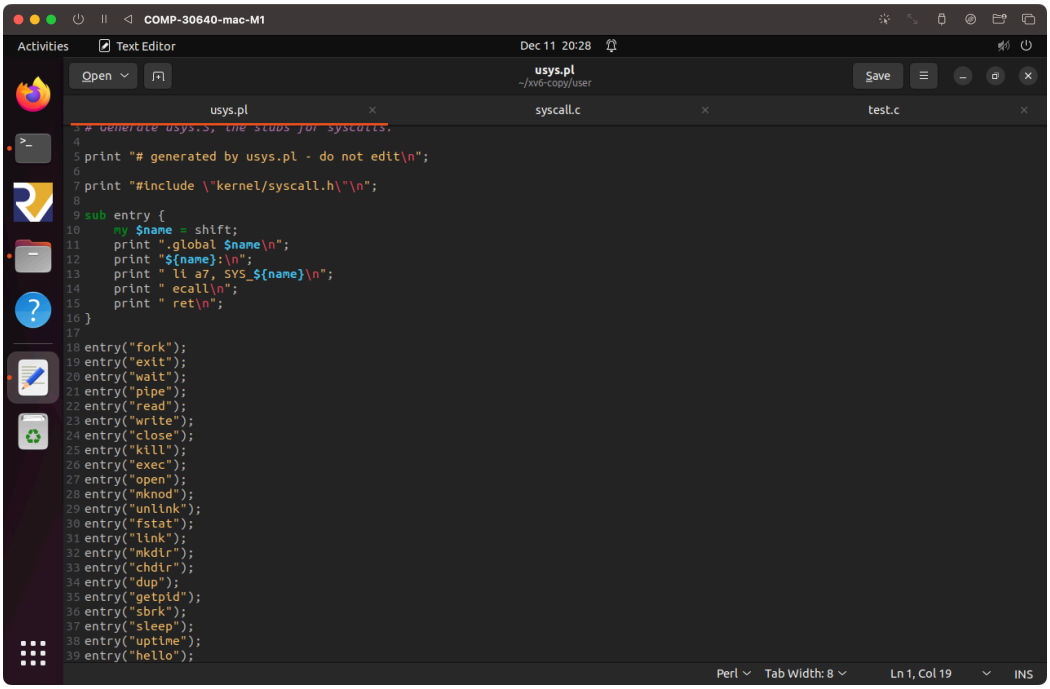

Figure 2: usys.pl file updated to register system call handler.

The test.c utility program invokes a new system call to obtain the Student ID returned by the `sys_hello()` function. As this program is written in C the printf function is implemented to display text “My Student ID is:” and the `%d` acts as a placeholder for the returned integer. As xv6-riscv does not have a standard stdio.h file the new system call handler function must be declared explicitly as an external function as is displayed in line 8 of Figure 3.

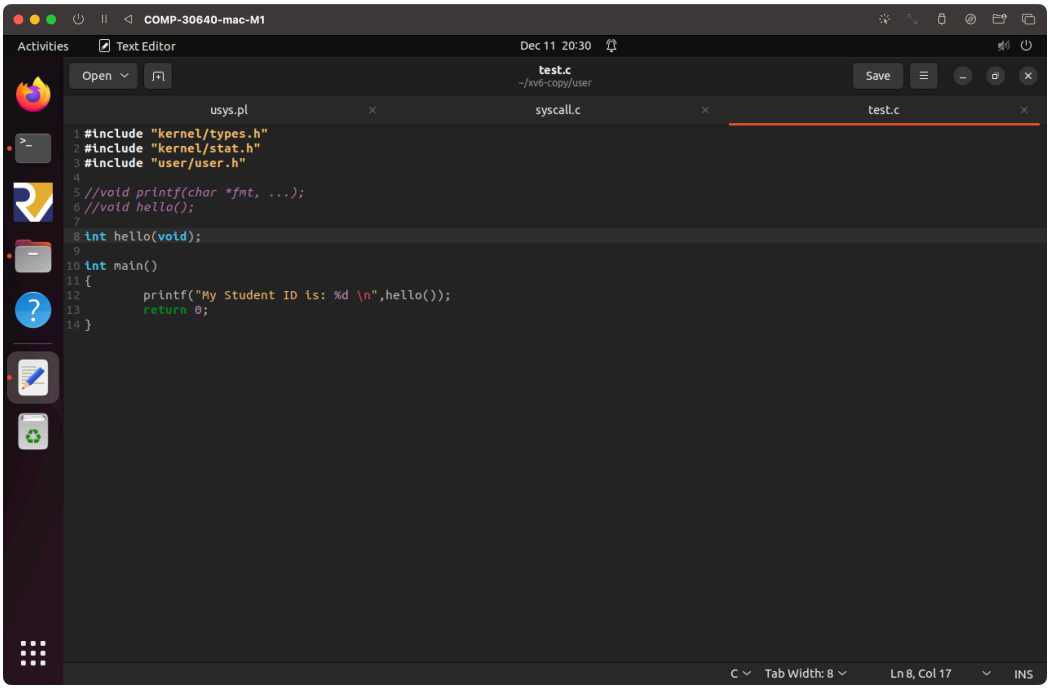

Figure 3: test.c system call handler function with explicit declaration.

Figure 4: Output of new system call function in xv6-copy folder.

Question 2

The downloader.py python program provided has been updated to include the additional functionality of processing simultaneous downloads by implementing threading. As is the case with the files in Question 1 of this assignment submission the updated python file has been included and an extract which highlights the threading implantation is displayed below:

```
for i in img_urls:
   threads.append(threading.Thread(target=download_image,args=(i,)))
for i in threads:
    i.start()
for i in threads:
    i.join()
```

The first for loop cycles through the images contained in the given array and creates a thread with the image downloaded function as its target and a single image from the array as its argument. These threads are then appended to an empty array allowing them to be held for the second for loop which begins each thread by calling the `.start()` method on all items in the list. Finally the `.join()` method is implemented to make the main thread wait until the newly created threads have executed and links the downloaded images back together.

An elementary script was also developed as a control to downloaded the images individually without implementing threading. Both downloader files were tested five times with the recorded results being displayed in Table 1:

Table 1: Results recorded from testing image downloader programs

(seconds)	Test #1	Test #2	Test #3	Test #4	Test #5	Average
No Threading	20.44	19.14	22.91	15.47	20.86	19.76
With Threading	9.78	11.52	12.05	13.14	17.47	12.79

Question 3

To compose a bash shell script which collects information from the /etc/passwd file which contains system information and concatenates the result with the paths to all world readable files on the system the following command is implemented:

```
{ cat /etc/passwd & find $PWD -perm -666; } > report.txt
```

The cat command takes the file as an argument and reads its contents. The find command searches for files by permission with the -perm condition of 666 translation to the binary value representing files which are writable for the owner, group and users of the file system. $PWD denotes that the full path name should be returned and the & operator concatenates the results of the cat and find commands. The curly braces containing these commands target a report which is not already on the file system, hence, report.txt is created to include the contents of the bash shell script. The execution and result of this expression is displayed in Figure 5. The full report.txt file is included as part of the assignment submission.

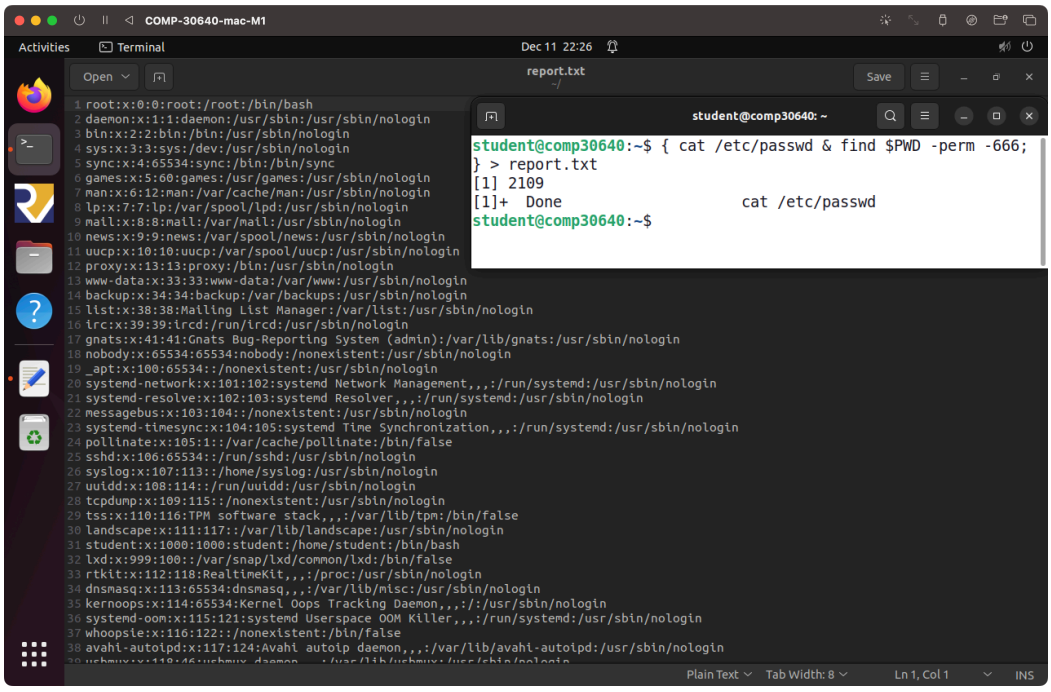

Figure 5: Execution and result of the bash shell script command.