CAOS ASSIGNMENT 1: SYSTEM CALL IMPLEMENTATION

-Parth Singh 2018356

**1🡪 Compiling the source code (assignment 0)**

The linux 3.16.71 source code is downloaded and it's extracted.

* cd to that directory
* Then 'cp /boot/config-$(**uname -r**) .config' to copy the kernels .config file
* Then **menuconfig** to configure it
* After that’s sudo make -j 6 (we add sudo in from off all make commands)
* **Sudo make modules\_install** -j 6 (giving 6 processors)
* **Sudo make install -j 6** (giving 6 processors)
* The kernel and modules are compiled.
* Then we have to **update-initramfs -c -k 3.16.71** to update the kernel.

**2🡪 Writing a System Call**

* Within our directory linux-3.16.71 we make a new directory and add the system call files there.
  + Make a new file **sh\_task\_info.c** - write\_to\_file function takes file\* and char \* as arguments and uses **write()** to write data in it.
  + In **sys\_sh\_task\_info(),**a task\_struct pointer task and struct file pointer file are created.
  + For each task, it's attributes are printed on the console and simultaneously written to the file.
  + A header file for it, sh\_task\_info.h
  + A Makefile for compiling the above.
* This new directory has to be added to the Linux-3.16.71. Then change the line **core -y += kernel/ mm/ fs/ ipc/ security/ crypto/ block/' to 'core -y += kernel/ mm/ fs/ ipc/ security/ crypto/ block/ hello/**
* New pid for the new system call is added in arch/x86/syscalls/syscall\_64.tbl. We added it at **318**.
* Then, the function was included in the **syscalls.h** in included.
* Then we compiled it **again** .
* The system is restarted.

**3🡪 Testing it with sample code**

* The file taskinfo.c is the sample test code which compares the process IDs by using syscall.
* **If input process ID and the updated pid 318 matches**
* Then it shows there is **no error** in the given pid and the filename **sh\_task\_info()** is executed correctly.