HW2 REPORT

Q2. Track system calls and library calls with File IO

Write a program that can do the following and show strace, ltrace and perf stat output:

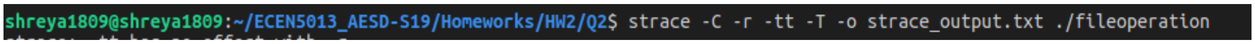
- Print to standard out an interesting string using printf
- Create a file
- Modify the permissions of the file to be read/write
- Open the file (for writing)
- Write a character to the file
- Close the file
- Open the file (in append mode)
- Dynamically allocate an array of memory
- Read an input string from the command line and write to the string to the allocated array
- Write the string to the file
- Flush file output
- Close the file
- Open the file (for reading)
- Read a single character (getc)
- Read a string of characters (gets)
- Close the file
- Free the memory

Show all 3 outputs (ltrace, strace, perf) in your report.

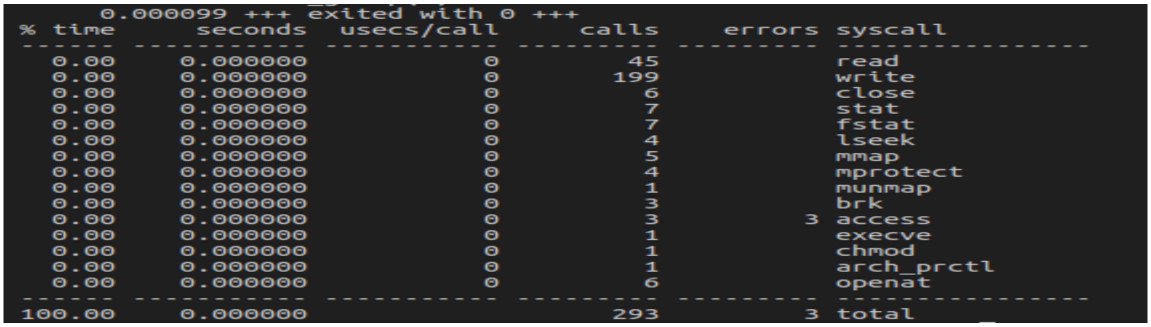
Ans: The main program is fileoperation.c with its functions in another c file called helperfunction.c. The logs of the three commands are given in strace_output.txt ,ltrace_output.txt and perf_output.txt respectively. All the files pertaining to Q2 is in a folder called Q2 in my git repo/Homeworks/HW2/Q2

1. **Strace:** It intercepts and records the system calls which are called by a process and the signals which are received by a process. The name of each system call, its arguments and its return value are printed on standard error or to the file specified with the -o option. (ref: man strace). In our case the entire log of the strace is given in the file strace_output.txt in Q2 folder.

Command:



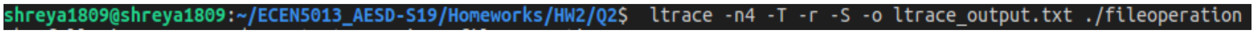
```
shreya1809@shreya1809:~/ECEN5013_AESD-S19/Homeworks/HW2/Q2$ strace -C -r -tt -T -o strace_output.txt ./fileoperation
```



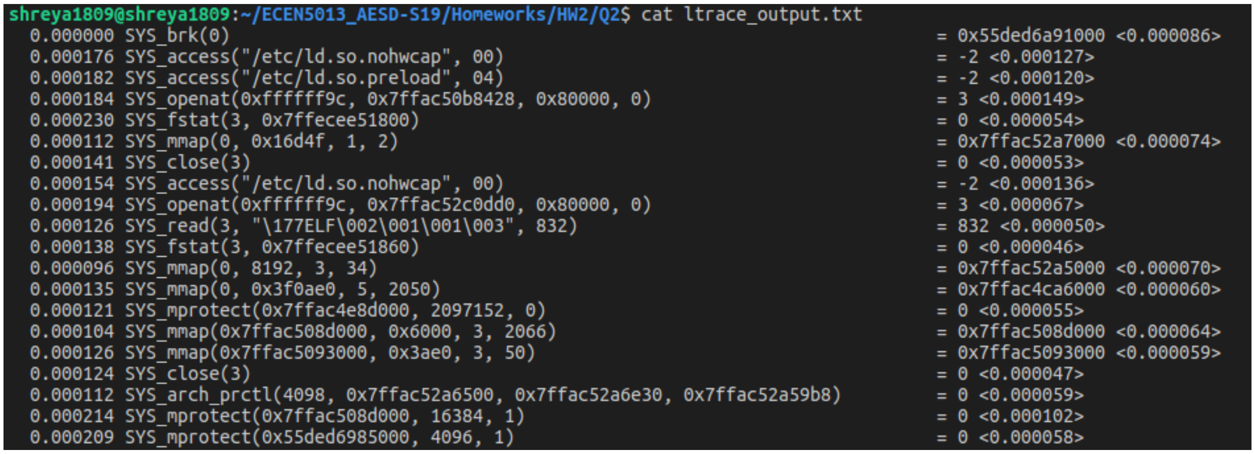
```
      0.000099 +++ exited with 0 +++
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
  0.00    0.000000           0        45           read
  0.00    0.000000           0       199           write
  0.00    0.000000           0         6           close
  0.00    0.000000           0         7           stat
  0.00    0.000000           0         7           fstat
  0.00    0.000000           0         4           lseek
  0.00    0.000000           0         5           mmap
  0.00    0.000000           0         4           mprotect
  0.00    0.000000           0         1           munmap
  0.00    0.000000           0         3           brk
  0.00    0.000000           0         3         3 access
  0.00    0.000000           0         1           execve
  0.00    0.000000           0         1           chmod
  0.00    0.000000           0         1           arch_prctl
  0.00    0.000000           0         6           openat
------ ----------- ----------- --------- --------- ----------------
100.00    0.000000                   293         3 total
```

2. **Ltrace:** It intercepts and records the dynamic library calls which are called by the executed process and the signals which are received by that process. It can also intercept and print the system calls executed by the program. (ref: man ltrace). In out case the entire log of the ltrace is given in the file ltrace_output.txt in Q2 folder.

Command:



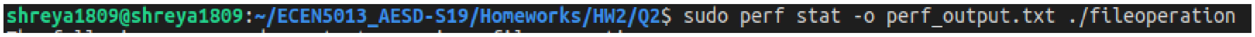
```
shreya1809@shreya1809:~/ECEN5013_AESD-S19/Homeworks/HW2/Q2$  ltrace -n4 -T -r -S -o ltrace_output.txt ./fileoperation
```



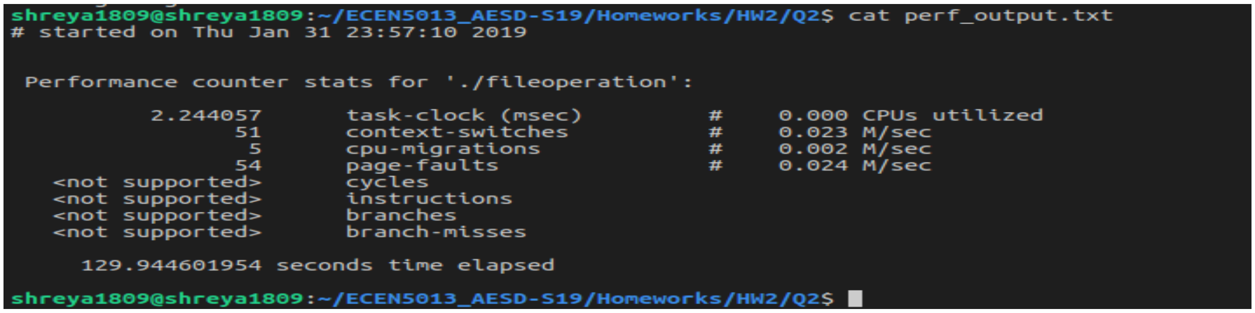
```
shreya1809@shreya1809:~/ECEN5013_AESD-S19/Homeworks/HW2/Q2$ cat ltrace_output.txt
  0.000000 SYS_brk(0)                                                              = 0x55ded6a91000 <0.000086>
  0.000176 SYS_access("/etc/ld.so.nohwcap", 00)                                    = -2 <0.000127>
  0.000182 SYS_access("/etc/ld.so.preload", 04)                                    = -2 <0.000120>
  0.000184 SYS_openat(0xffffff9c, 0x7ffac50b8428, 0x80000, 0)                      = 3 <0.000149>
  0.000230 SYS_fstat(3, 0x7ffecee51800)                                            = 0 <0.000054>
  0.000112 SYS_mmap(0, 0x16d4f, 1, 2)                                              = 0x7ffac52a7000 <0.000074>
  0.000141 SYS_close(3)                                                            = 0 <0.000053>
  0.000154 SYS_access("/etc/ld.so.nohwcap", 00)                                    = -2 <0.000136>
  0.000194 SYS_openat(0xffffff9c, 0x7ffac52c0dd0, 0x80000, 0)                      = 3 <0.000067>
  0.000126 SYS_read(3, "\177ELF\002\001\001\003", 832)                             = 832 <0.000050>
  0.000138 SYS_fstat(3, 0x7ffecee51860)                                            = 0 <0.000046>
  0.000096 SYS_mmap(0, 8192, 3, 34)                                                = 0x7ffac52a5000 <0.000070>
  0.000135 SYS_mmap(0, 0x3f0ae0, 5, 2050)                                          = 0x7ffac4ca6000 <0.000060>
  0.000121 SYS_mprotect(0x7ffac4e8d000, 2097152, 0)                                = 0 <0.000055>
  0.000104 SYS_mmap(0x7ffac508d000, 0x6000, 3, 2066)                               = 0x7ffac508d000 <0.000064>
  0.000126 SYS_mmap(0x7ffac5093000, 0x3ae0, 3, 50)                                 = 0x7ffac5093000 <0.000059>
  0.000124 SYS_close(3)                                                            = 0 <0.000047>
  0.000112 SYS_arch_prctl(4098, 0x7ffac52a6500, 0x7ffac52a6e30, 0x7ffac52a59b8)    = 0 <0.000059>
  0.000214 SYS_mprotect(0x7ffac508d000, 16384, 1)                                  = 0 <0.000102>
  0.000209 SYS_mprotect(0x55ded6985000, 4096, 1)                                   = 0 <0.000058>
```

3. **Perf** : This command runs a command and gathers performance counter statistics from it. The perf stat log is given in the file perf_output.txt in the folder Q2.

Command:



```
shreya1809@shreya1809:~/ECEN5013_AESD-S19/Homeworks/HW2/Q2$ sudo perf stat -o perf_output.txt ./fileoperation
```



```
shreya1809@shreya1809:~/ECEN5013_AESD-S19/Homeworks/HW2/Q2$ cat perf_output.txt
# started on Thu Jan 31 23:57:10 2019


 Performance counter stats for './fileoperation':

          2.244057      task-clock (msec)         #    0.000 CPUs utilized
                51      context-switches          #    0.023 M/sec
                 5      cpu-migrations            #    0.002 M/sec
                54      page-faults               #    0.024 M/sec
   <not supported>      cycles
   <not supported>      instructions
   <not supported>      branches
   <not supported>      branch-misses

     129.944601954 seconds time elapsed

shreya1809@shreya1809:~/ECEN5013_AESD-S19/Homeworks/HW2/Q2$ █
```

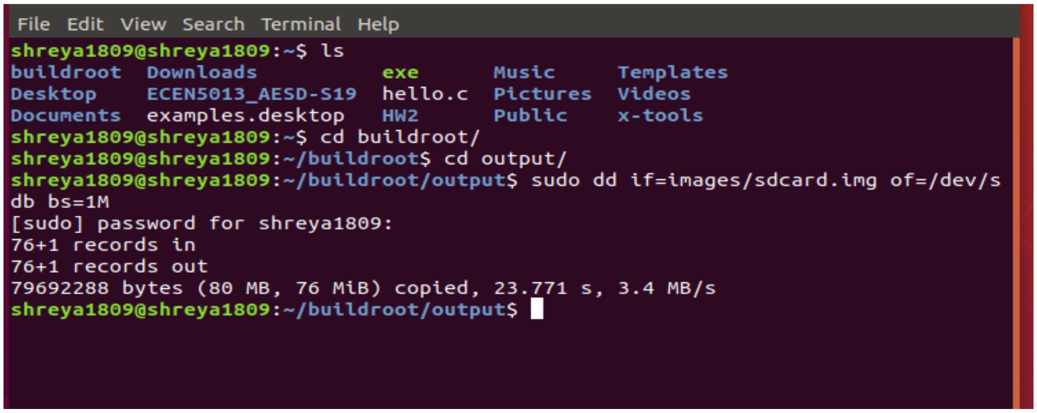
Q3. Setup Buildroot, then build and boot a Beaglebone Green (BBG) Linux image.

Reference: https://www.youtube.com/watch?v=QLzGKIx9Mz8

The steps involved in this are as follows:

1. You need to install the dependencies for buildroot. Check for all the mandatory packages that need to be installed (sudo apt-get install sed make binutils gcc g++ bash patch \ gzip bzip2 perl tar cpio python unzip rsync wget libncurses-dev)

2. Clone buildroot from the link : https://github.com/buildroot/buildroot this clones the latest version of buildroot.
3. Go to cd buildroot/configs, do 'ls' and see if Beaglebone_defconfig is present
4. Go back to buildroot directory and do "sudo make menuconfig" to open the menu for configuration options
5. In the menuconfig, go to 'load' option on the bottom of the screen using tab and press enter.
6. Enter the name of the configuration file "Beaglebone_defconfig" so that now the path looks like : 'home/shreya1809/buildroot/configs/Beaglebone_defconfig' and press 'ok'.
7. To change the welcome message got to 'System Configuration' from the main menu and press enter. Then scroll down using the down arrow key to the 'system banner'. Press enter and enter the string you wish to make the welcome message and press 'ok'.
8. To enable password scroll down to the "Enable root Login with password" checkbox and press spacebar and change the root password.
9. After doing all the necessary changes as required in the menuconfig, go to the 'save' option at the bottom of the screen. This will open a dialogue box saying "Enter a name to which this configuration should be saved as alternate". We save this as the default name of configuration. The path now would look like this: 'home/shreya1809/buildroot/.config' instead of : 'home/shreya1809/buildroot/configs/Beaglebone_defconfig'
10. Exit the menuconfig. And do 'sudo make menuconfig' to build the configuration. This might take several hours.
11. This building process using buildroot will automatically generate the sdcard image that we need to burn in our SD card. To check the image got to buildroot/output/images -> sdcard.img
12. Insert you sd card and detect it in the virtual machine. https://www.howtogeek.com/howto/31726/mount-usb-devices-in-virtualbox-with-ubuntu/
13. Insert your sd card using Micro sd card reader or whichever way is comfortable for you. Format the sdcard using diskpart in windows (tutorial : https://www.windowscentral.com/how-clean-and-format-storage-drive-using-diskpart-windows-10) or gparted in ubuntu. Remove all the partitions and create a partition.
14. Got to buildroot/output/images and Write the sdcard.img to the sd card using the following command : "sudo dd if=sdcard.img of=/dev/sdb bs=1M"
15. Around 79MB of data will be written to the sdcard as shown below:



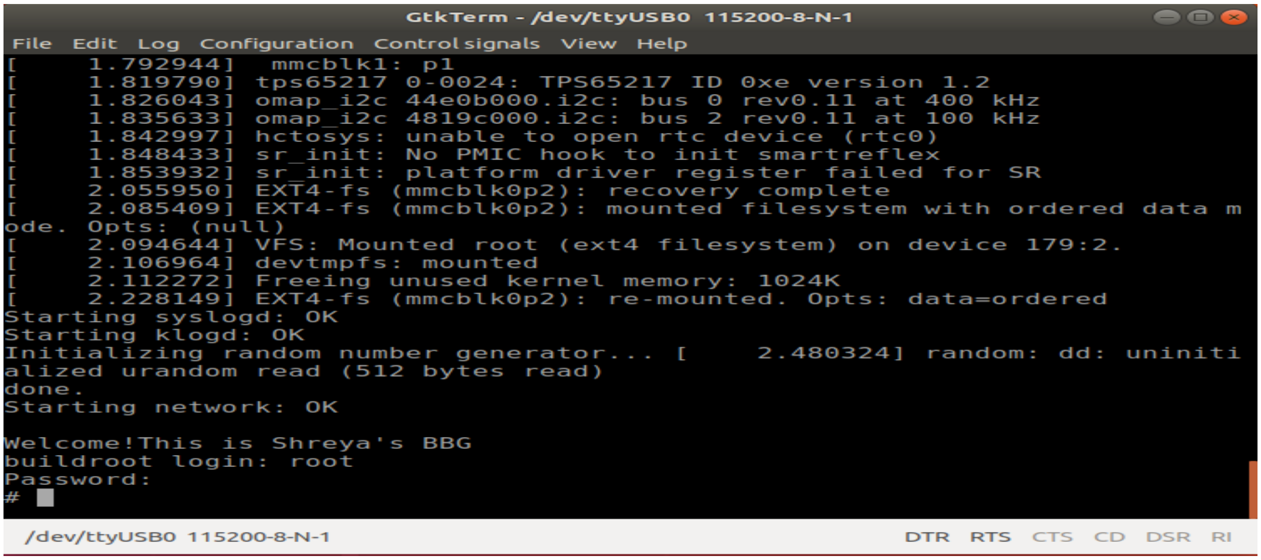
```
shreya1809@shreya1809:~$ ls
buildroot  Downloads          exe      Music     Templates
Desktop    ECEN5013_AESD-S19  hello.c  Pictures  Videos
Documents  examples.desktop   HW2      Public    x-tools
shreya1809@shreya1809:~$ cd buildroot/
shreya1809@shreya1809:~/buildroot$ cd output/
shreya1809@shreya1809:~/buildroot/output$ sudo dd if=images/sdcard.img of=/dev/s
db bs=1M
[sudo] password for shreya1809:
76+1 records in
76+1 records out
79692288 bytes (80 MB, 76 MiB) copied, 23.771 s, 3.4 MB/s
shreya1809@shreya1809:~/buildroot/output$
```

16. Remove the sdcard and insert it into BBG. Connect the BBG to the laptop using the FTDI/USB cable. Black wire to J1 on BBG.
17. Install your favorite emulator. I will be using screen and gtkterm interchangeably.

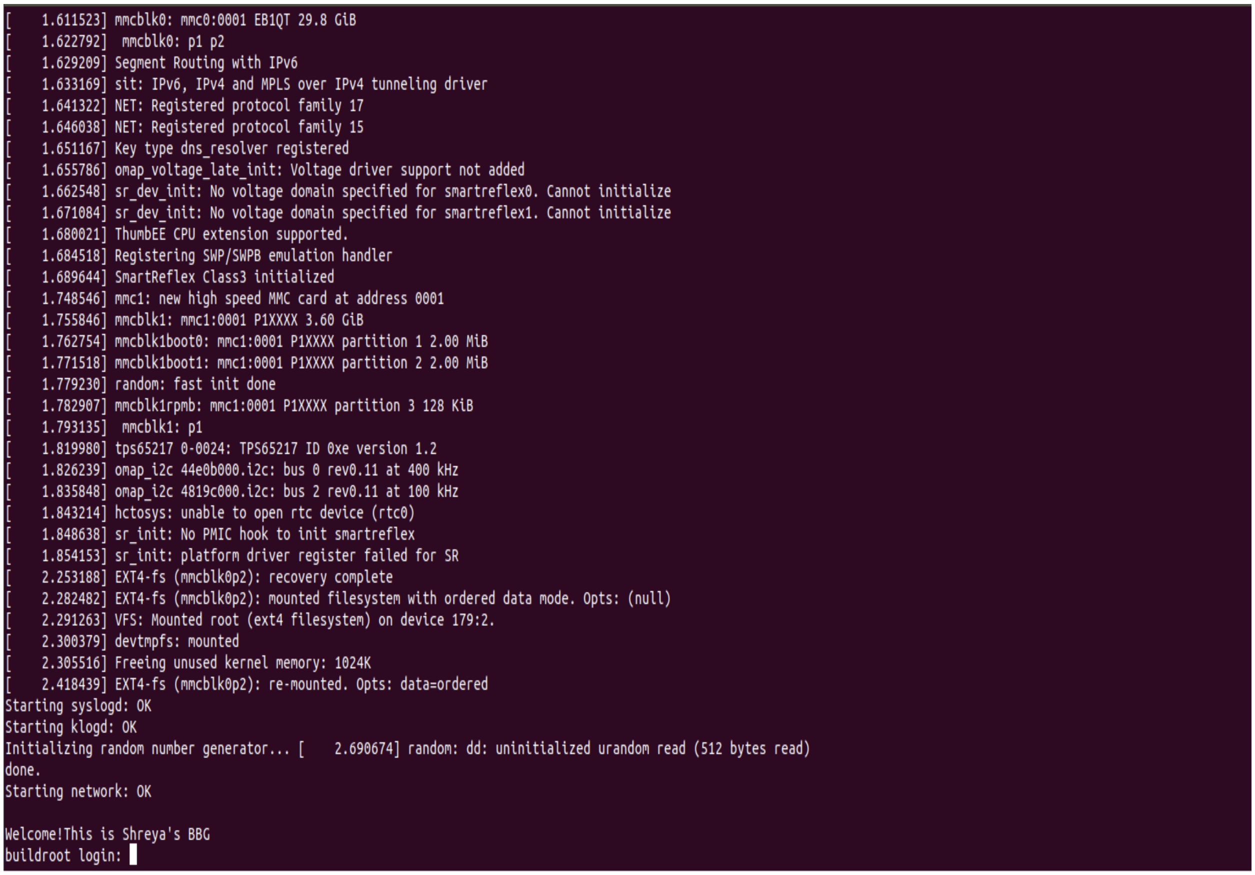
To run screen the command is : “$ sudo screen /dev/ttyUSB0 115200 “
To run gtkterm, install using apt-get intall gtkterm and run it using the command:“$ sudo gtkterm -s 115200 -p /dev/ttyUSB0 ”

18. Run “screen” and after that give power to the BBG. It should show boot up logs followed by the welcome message as below:

Welcome screen with my login name

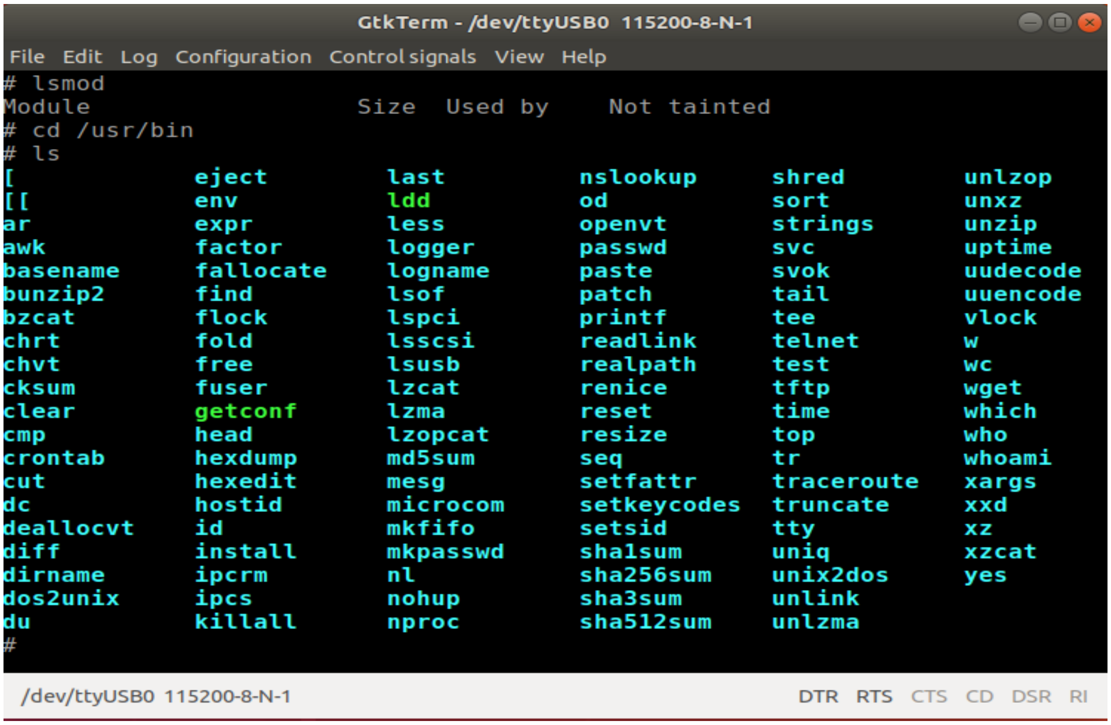

GtkTerm - /dev/ttyUSB0 115200-8-N-1
File Edit Log Configuration Control signals View Help
/dev/ttyUSB0 115200-8-N-1
DTR RTS CTS CD DSR RI

```
[      1.792944]  mmcblk1: p1
[      1.819790] tps65217 0-0024: TPS65217 ID 0xe version 1.2
[      1.826043] omap_i2c 44e0b000.i2c: bus 0 rev0.11 at 400 kHz
[      1.835633] omap_i2c 4819c000.i2c: bus 2 rev0.11 at 100 kHz
[      1.842997] hctosys: unable to open rtc device (rtc0)
[      1.848433] sr_init: No PMIC hook to init smartreflex
[      1.853932] sr_init: platform driver register failed for SR
[      2.055950] EXT4-fs (mmcblk0p2): recovery complete
[      2.085409] EXT4-fs (mmcblk0p2): mounted filesystem with ordered data m
ode. Opts: (null)
[      2.094644] VFS: Mounted root (ext4 filesystem) on device 179:2.
[      2.106964] devtmpfs: mounted
[      2.112272] Freeing unused kernel memory: 1024K
[      2.228149] EXT4-fs (mmcblk0p2): re-mounted. Opts: data=ordered
Starting syslogd: OK
Starting klogd: OK
Initializing random number generator... [    2.480324] random: dd: uniniti
alized urandom read (512 bytes read)
done.
Starting network: OK

Welcome!This is Shreya's BBG
buildroot login: root
Password:
# 
```

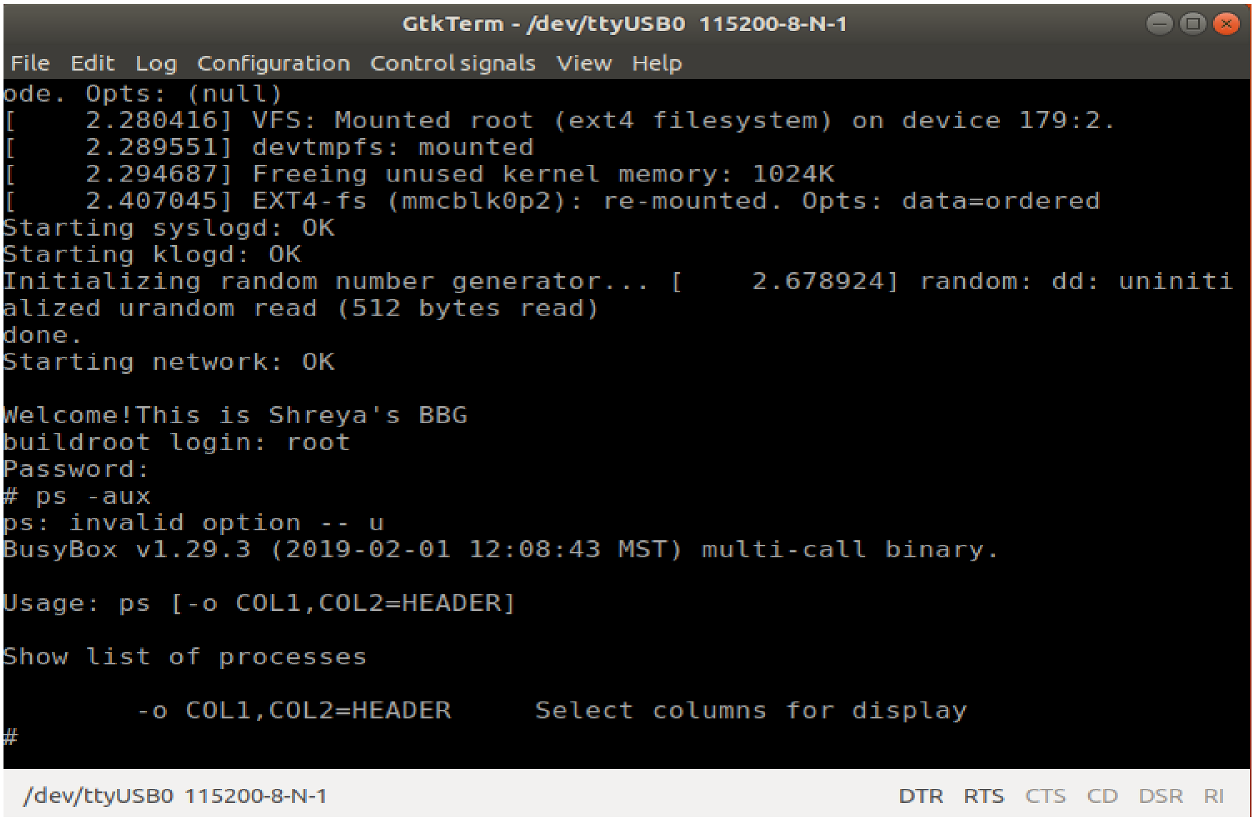


```
[    1.611523] mmcblk0: mmc0:0001 EB1QT 29.8 GiB
[    1.622792]  mmcblk0: p1 p2
[    1.629209] Segment Routing with IPv6
[    1.633169] sit: IPv6, IPv4 and MPLS over IPv4 tunneling driver
[    1.641322] NET: Registered protocol family 17
[    1.646038] NET: Registered protocol family 15
[    1.651167] Key type dns_resolver registered
[    1.655786] omap_voltage_late_init: Voltage driver support not added
[    1.662548] sr_dev_init: No voltage domain specified for smartreflex0. Cannot initialize
[    1.671084] sr_dev_init: No voltage domain specified for smartreflex1. Cannot initialize
[    1.680021] ThumbEE CPU extension supported.
[    1.684518] Registering SWP/SWPB emulation handler
[    1.689644] SmartReflex Class3 initialized
[    1.748546] mmc1: new high speed MMC card at address 0001
[    1.755846] mmcblk1: mmc1:0001 P1XXXX 3.60 GiB
[    1.762754] mmcblk1boot0: mmc1:0001 P1XXXX partition 1 2.00 MiB
[    1.771518] mmcblk1boot1: mmc1:0001 P1XXXX partition 2 2.00 MiB
[    1.779230] random: fast init done
[    1.782907] mmcblk1rpmb: mmc1:0001 P1XXXX partition 3 128 KiB
[    1.793135]  mmcblk1: p1
[    1.819980] tps65217 0-0024: TPS65217 ID 0xe version 1.2
[    1.826239] omap_i2c 44e0b000.i2c: bus 0 rev0.11 at 400 kHz
[    1.835848] omap_i2c 4819c000.i2c: bus 2 rev0.11 at 100 kHz
[    1.843214] hctosys: unable to open rtc device (rtc0)
[    1.848638] sr_init: No PMIC hook to init smartreflex
[    1.854153] sr_init: platform driver register failed for SR
[    2.253188] EXT4-fs (mmcblk0p2): recovery complete
[    2.282482] EXT4-fs (mmcblk0p2): mounted filesystem with ordered data mode. Opts: (null)
[    2.291263] VFS: Mounted root (ext4 filesystem) on device 179:2.
[    2.300379] devtmpfs: mounted
[    2.305516] Freeing unused kernel memory: 1024K
[    2.418439] EXT4-fs (mmcblk0p2): re-mounted. Opts: data=ordered
Starting syslogd: OK
Starting klogd: OK
Initializing random number generator... [    2.690674] random: dd: uninitialized urandom read (512 bytes read)
done.
Starting network: OK

Welcome!This is Shreya's BBG
buildroot login: 
```

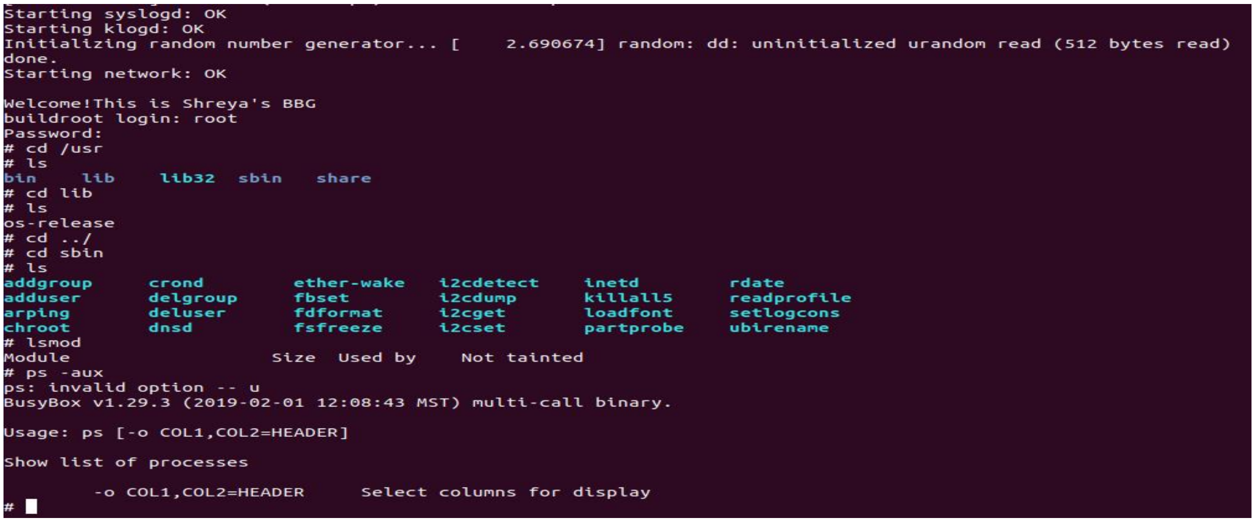
Commands executed: ls,lsmod and ps -aux


GtkTerm - /dev/ttyUSB0 115200-8-N-1
File Edit Log Configuration Control signals View Help
/dev/ttyUSB0 115200-8-N-1
DTR RTS CTS CD DSR RI

```
# lsmod
Module                  Size  Used by    Not tainted
# cd /usr/bin
# ls
[            eject        last         nslookup     shred        unlzop
[[           env          ldd          od           sort         unxz
ar           expr         less         openvt       strings      unzip
awk          factor       logger       passwd       svc          uptime
basename     fallocate    logname      paste        svok         uudecode
bunzip2      find         lsof         patch        tail         uuencode
bzcat        flock        lspci        printf       tee          vlock
chrt         fold         lsscsi       readlink     telnet       w
chvt         free         lsusb        realpath     test         wc
cksum        fuser        lzcat        renice       tftp         wget
clear        getconf      lzma         reset        time         which
cmp          head         lzopcat      resize       top          who
crontab      hexdump      md5sum       seq          tr           whoami
cut          hexedit      mesg         setfattr     traceroute   xargs
dc           hostid       microcom     setkeycodes  truncate     xxd
deallocvt    id           mkfifo       setsid       tty          xz
diff         install      mkpasswd     sha1sum      uniq         xzcat
dirname      ipcrm        nl           sha256sum    unix2dos     yes
dos2unix     ipcs         nohup        sha3sum      unlink
du           killall      nproc        sha512sum    unlzma
#
```


GtkTerm - /dev/ttyUSB0 115200-8-N-1
File Edit Log Configuration Control signals View Help
/dev/ttyUSB0 115200-8-N-1
DTR RTS CTS CD DSR RI

```
ode. Opts: (null)
[    2.280416] VFS: Mounted root (ext4 filesystem) on device 179:2.
[    2.289551] devtmpfs: mounted
[    2.294687] Freeing unused kernel memory: 1024K
[    2.407045] EXT4-fs (mmcblk0p2): re-mounted. Opts: data=ordered
Starting syslogd: OK
Starting klogd: OK
Initializing random number generator... [    2.678924] random: dd: uniniti
alized urandom read (512 bytes read)
done.
Starting network: OK

Welcome!This is Shreya's BBG
buildroot login: root
Password:
# ps -aux
ps: invalid option -- u
BusyBox v1.29.3 (2019-02-01 12:08:43 MST) multi-call binary.

Usage: ps [-o COL1,COL2=HEADER]

Show list of processes

        -o COL1,COL2=HEADER     Select columns for display
#
```



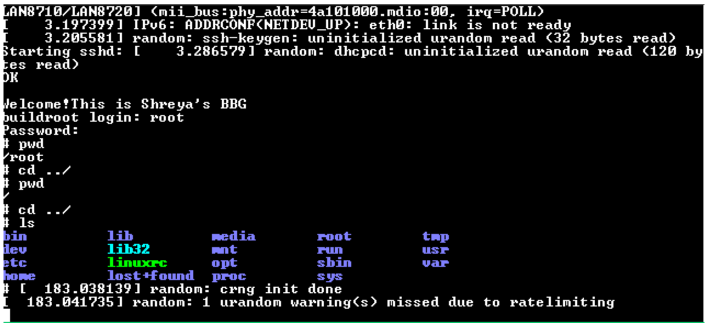
```
Starting syslogd: OK
Starting klogd: OK
Initializing random number generator... [    2.690674] random: dd: uninitialized urandom read (512 bytes read)
done.
Starting network: OK

Welcome!This is Shreya's BBG
buildroot login: root
Password:
# cd /usr
# ls
bin     lib     lib32   sbin    share
# cd lib
# ls
os-release
# cd ../
# cd sbin
# ls
addgroup      crond         ether-wake    i2cdetect     inetd         rdate
adduser       delgroup      fbset         i2cdump       killall5      readprofile
arping        deluser       fdformat      i2cget        loadfont      setlogcons
chroot        dnsd          fsfreeze      i2cset        partprobe     ubirename
# lsmod
Module                  Size  Used by    Not tainted
# ps -aux
ps: invalid option -- u
BusyBox v1.29.3 (2019-02-01 12:08:43 MST) multi-call binary.

Usage: ps [-o COL1,COL2=HEADER]

Show list of processes

        -o COL1,COL2=HEADER     Select columns for display
# 
```

Q4. porting file I/O program to BBG

Ans: Steps for porting file to BBG

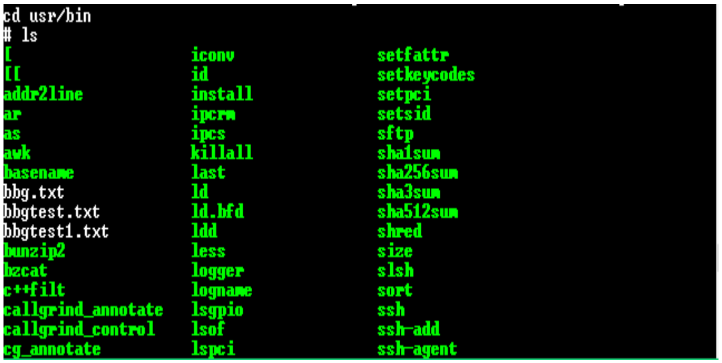
1. For porting the file to the BBG, it first needs to be cross compiled on the host machine. Hence the cross compiler has been changed in the makefile of Q4 from Q2. In Q2 it was “CC = gcc” and here it is “CC = /home/shreya1809//buildroot/output/host/usr/bim/arm-buildroot-linux-uclibcgnueabihf-gcc”.The program is run and the executable is saved as “fileoperation” in my case.
2. The executable is then saved in a directory created by the name of “root-overlay” in buildroot/board/Beaglebone as given in the question. Hence the new path to the cross compiled executable is : “/buildroot/board/beaglebone/root-overlay/usr/bin”
3. Made changes in the menuconfigs (some of them are already there by default):
    a. In kernel -> select ‘Linux Kernel’ , ‘Build a device tree Blob (DTB)’ , Linux kernel tools -> select ‘perf’ , ‘enable perf TUI’.
    b. In Build Options -> select ‘strip target binaries’, ‘Enable compiler cache’, ‘Use relative paths.
    c. In Toolchain -> select ‘Enable WCHAR support’ , ‘Compile and install uclibc utilities’ , ‘Enable MMU support’ .
    d. In System Configuration -> select ‘Enable root login with password’ , ‘Run a getty after boot’ , ‘remount root filesystem read-write during boot’ , ‘Purge unwanted locales’
    e. In target packages -> Dubugging,profiling and benchmark -> select ‘strace’ , ‘ltrace’.
    f. In target packages -> Shell and utilities -> select ‘screen’ , ‘sudo’, ‘zsh’ , ‘file’
    g. In target packages -> System tools -> select ‘cpuload’ , ‘daemon’
    h. In Filesystem images -> select ‘ext2/3/4 root filesystem and select ext4 from the options.
    i. In Filesystem images -> exact size - > default is given as 60M, increase it to 320M.
    j. In Filesystem images -> select ‘tar the root filesystem’
    k. In bootloaders -> select ‘U-Boot’ , ‘U-Boot needs dtc’ , ‘ Install U-Boot SPL binary image’
    l. In Host Utilities - > select host ‘dosftools’ , ‘genimage’ , ‘mfgtools’ , ‘mtools’.
4. sudo make menuconfig. This might take sometime.

5. Repeat the steps in Q4. Format the SD card, delete partitions, burn the new image into the sd card, load the SD card in the BBG. Connect the FTDI/USB cable to laptop with the black wire in J1 of BBG.
6. Open Tera Term/Screen/gtkterm/putty whatever is more convenient. I chose Tera Term. Select serial and put the correct COM port by checking in the device manager. Press okay. Go to setup from the menu bar and choose serial port. Give the speed as 115200. Press Ok. Give power to the BBG and see it boot up.
7. Do cd ../../ and ls

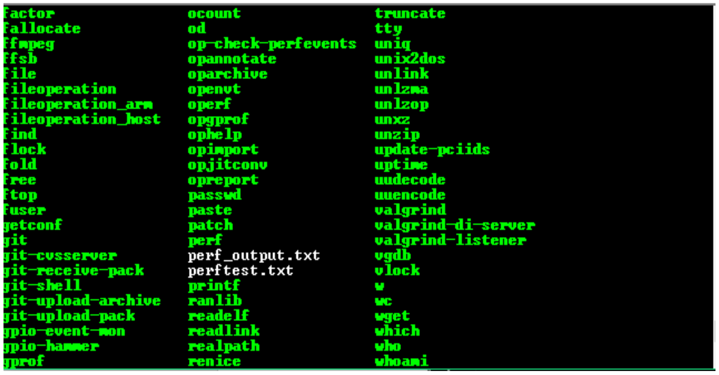


```
LAN8710/LAN8720] (mii_bus:phy_addr=4a101000.mdio:00, irq=POLL)
[    3.197399] IPv6: ADDRCONF(NETDEV_UP): eth0: link is not ready
[    3.205581] random: ssh-keygen: uninitialized urandom read (32 bytes read)
Starting sshd: [    3.286579] random: dhcpcd: uninitialized urandom read (120 by
tes read)
OK

Welcome!This is Shreya's BBG
buildroot login: root
Password:
# pwd
/root
# cd ../
# pwd
/
# cd ../
# ls
bin         lib         media       root        tmp
dev         lib32       mnt         run         usr
etc         linuxrc     opt         sbin        var
home        lost+found  proc        sys
# [  183.038139] random: crng init done
[  183.041735] random: 1 urandom warning(s) missed due to ratelimiting
```

8. cd usr/bin and then ls



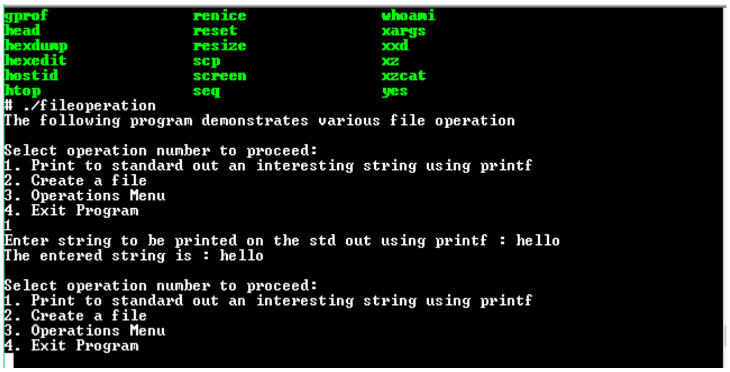
```
cd usr/bin
# ls
[                     iconv                 setfattr
[[                    id                    setkeycodes
addr2line             install               setpci
ar                    ipcrm                 setsid
as                    ipcs                  sftp
awk                   killall               sha1sum
basename              last                  sha256sum
bbg.txt               ld                    sha3sum
bbgtest.txt           ld.bfd                sha512sum
bbgtest1.txt          ldd                   shred
bunzip2               less                  size
bzcat                 logger                slsh
c++filt               logname               sort
callgrind_annotate    lsgpio                ssh
callgrind_control     lsof                  ssh-add
cg_annotate           lspci                 ssh-agent
```



```
factor                ocount                truncate
fallocate             od                    tty
ffmpeg                op-check-perfevents   uniq
ffsb                  opannotate            unix2dos
file                  oparchive             unlink
fileoperation         openvt                unlzma
fileoperation_arm     operf                 unlzop
fileoperation_host    opgprof               unxz
find                  ophelp                unzip
flock                 opimport              update-pciids
fold                  opjitconv             uptime
free                  opreport              uudecode
ftop                  passwd                uuencode
fuser                 paste                 valgrind
getconf               patch                 valgrind-di-server
git                   perf                  valgrind-listener
git-cvsserver         perf_output.txt       vgdb
git-receive-pack      perftest.txt          vlock
git-shell             printf                w
git-upload-archive    ranlib                wc
git-upload-pack       readelf               wget
gpio-event-mon        readlink              which
gpio-hammer           realpath              who
gprof                 renice                whoami
```

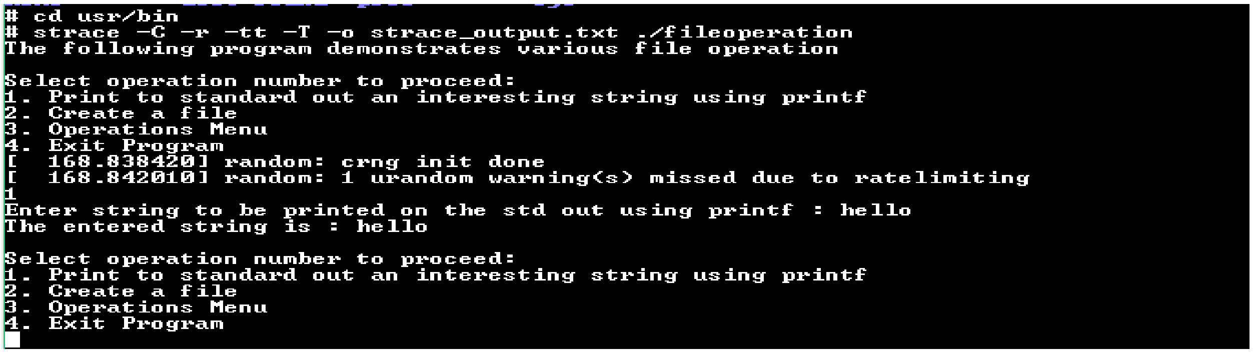
Observe that our executable called "fileoperation" is present in the list. Next we execute it.

9. ./fileoperation



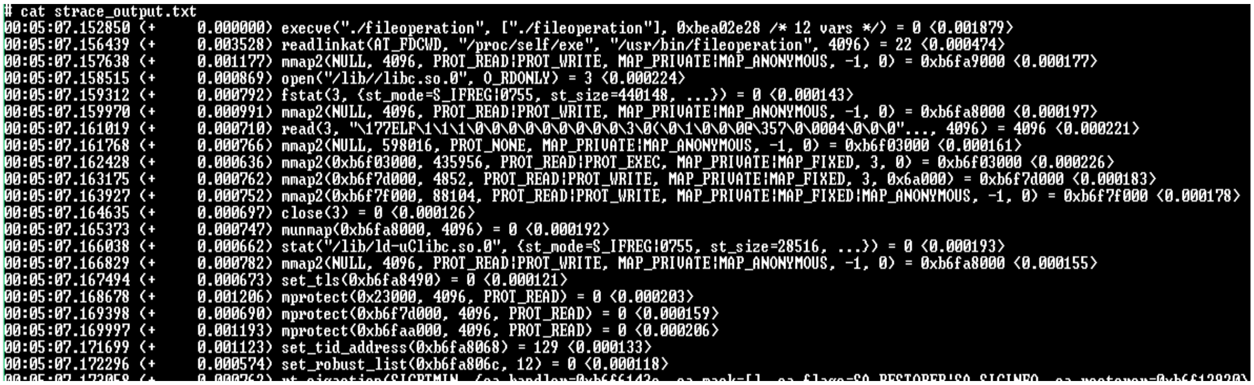
```
gprof                 renice               whoami
head                  reset                xargs
hexdump               resize               xxd
hexedit               scp                  xz
hostid                screen               xzcat
htop                  seq                  yes
# ./fileoperation
The following program demonstrates various file operation

Select operation number to proceed:
1. Print to standard out an interesting string using printf
2. Create a file
3. Operations Menu
4. Exit Program
1
Enter string to be printed on the std out using printf : hello
The entered string is : hello

Select operation number to proceed:
1. Print to standard out an interesting string using printf
2. Create a file
3. Operations Menu
4. Exit Program
```

Strace Output:



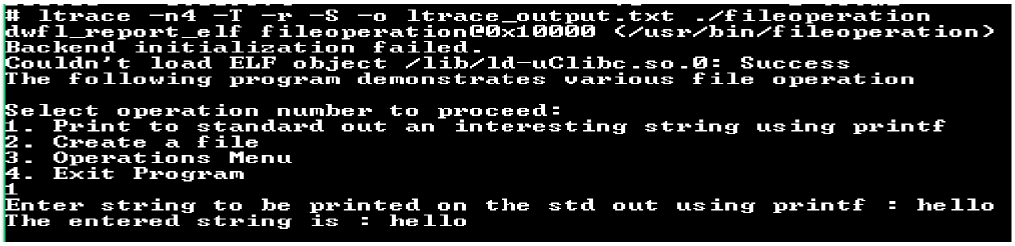
```
# cd usr/bin
# strace -C -r -tt -T -o strace_output.txt ./fileoperation
The following program demonstrates various file operation

Select operation number to proceed:
1. Print to standard out an interesting string using printf
2. Create a file
3. Operations Menu
4. Exit Program
[  168.838420] random: crng init done
[  168.842010] random: 1 urandom warning(s) missed due to ratelimiting
1
Enter string to be printed on the std out using printf : hello
The entered string is : hello

Select operation number to proceed:
1. Print to standard out an interesting string using printf
2. Create a file
3. Operations Menu
4. Exit Program
```

Cat strace_output.txt



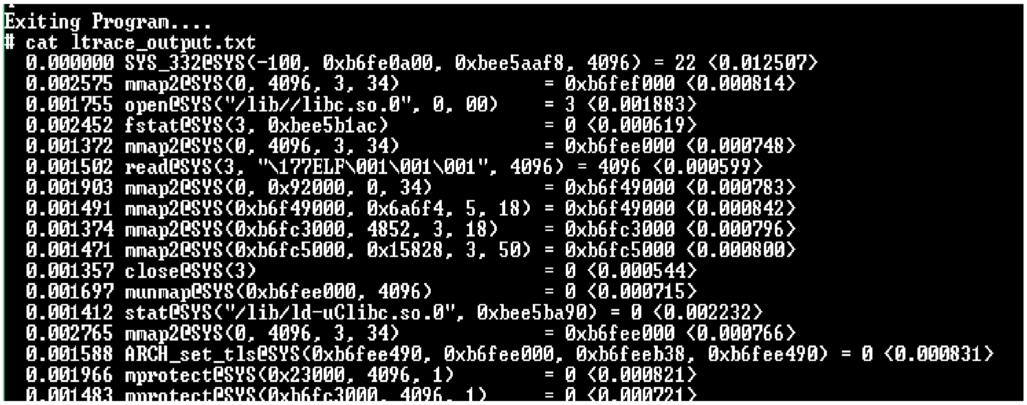
```
# cat strace_output.txt
00:05:07.152850 (+     0.000000) execve("./fileoperation", ["./fileoperation"], 0xbea02e28 /* 12 vars */) = 0 <0.001879>
00:05:07.156439 (+     0.003528) readlinkat(AT_FDCWD, "/proc/self/exe", "/usr/bin/fileoperation", 4096) = 22 <0.000474>
00:05:07.157638 (+     0.001177) mmap2(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0xb6fa9000 <0.000177>
00:05:07.158515 (+     0.000869) open("/lib//libc.so.0", O_RDONLY) = 3 <0.000224>
00:05:07.159312 (+     0.000792) fstat(3, {st_mode=S_IFREG|0755, st_size=440148, ...}) = 0 <0.000143>
00:05:07.159970 (+     0.000991) mmap2(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0xb6fa8000 <0.000197>
00:05:07.161019 (+     0.000710) read(3, "\177ELF\1\1\1\0\0\0\0\0\0\0\0\3\0(\0\1\0\0\0@\357\0\0004\0\0\0"..., 4096) = 4096 <0.000221>
00:05:07.161768 (+     0.000766) mmap2(NULL, 598016, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0xb6f03000 <0.000161>
00:05:07.162428 (+     0.000636) mmap2(0xb6f03000, 435956, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED, 3, 0) = 0xb6f03000 <0.000226>
00:05:07.163175 (+     0.000762) mmap2(0xb6f7d000, 4852, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED, 3, 0x6a000) = 0xb6f7d000 <0.000183>
00:05:07.163927 (+     0.000752) mmap2(0xb6f7f000, 88104, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0xb6f7f000 <0.000178>
00:05:07.164635 (+     0.000697) close(3) = 0 <0.000126>
00:05:07.165373 (+     0.000747) munmap(0xb6fa8000, 4096) = 0 <0.000192>
00:05:07.166038 (+     0.000662) stat("/lib/ld-uClibc.so.0", {st_mode=S_IFREG|0755, st_size=28516, ...}) = 0 <0.000193>
00:05:07.166829 (+     0.000782) mmap2(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0xb6fa8000 <0.000155>
00:05:07.167494 (+     0.000673) set_tls(0xb6fa8490) = 0 <0.000121>
00:05:07.168678 (+     0.001206) mprotect(0x23000, 4096, PROT_READ) = 0 <0.000203>
00:05:07.169398 (+     0.000690) mprotect(0xb6f7d000, 4096, PROT_READ) = 0 <0.000159>
00:05:07.169997 (+     0.001193) mprotect(0xb6faa000, 4096, PROT_READ) = 0 <0.000206>
00:05:07.171699 (+     0.001123) set_tid_address(0xb6fa8068) = 129 <0.000133>
00:05:07.172296 (+     0.000574) set_robust_list(0xb6fa806c, 12) = 0 <0.000118>
```

Ltrace Output:



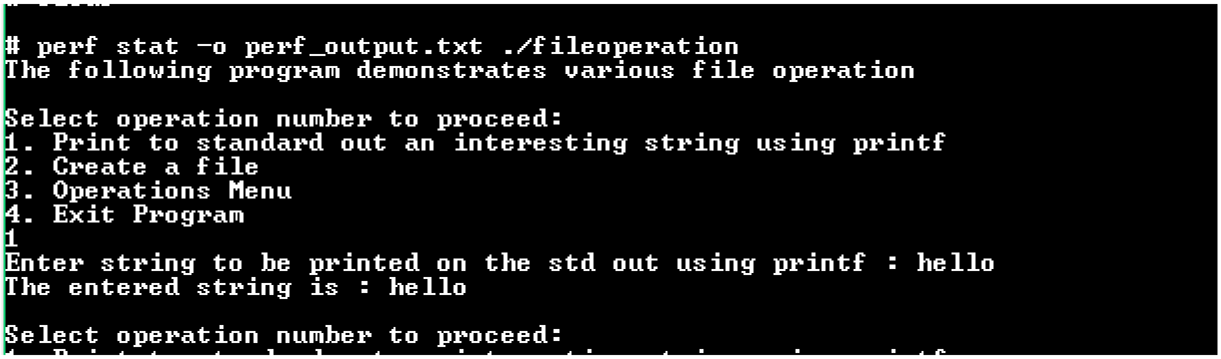
```
# ltrace -n4 -T -r -S -o ltrace_output.txt ./fileoperation
dwfl_report_elf fileoperation@0x10000 (/usr/bin/fileoperation)
Backend initialization failed.
Couldn't load ELF object /lib/ld-uClibc.so.0: Success
The following program demonstrates various file operation

Select operation number to proceed:
1. Print to standard out an interesting string using printf
2. Create a file
3. Operations Menu
4. Exit Program
1
Enter string to be printed on the std out using printf : hello
The entered string is : hello
```

Cat ltrace_output.txt



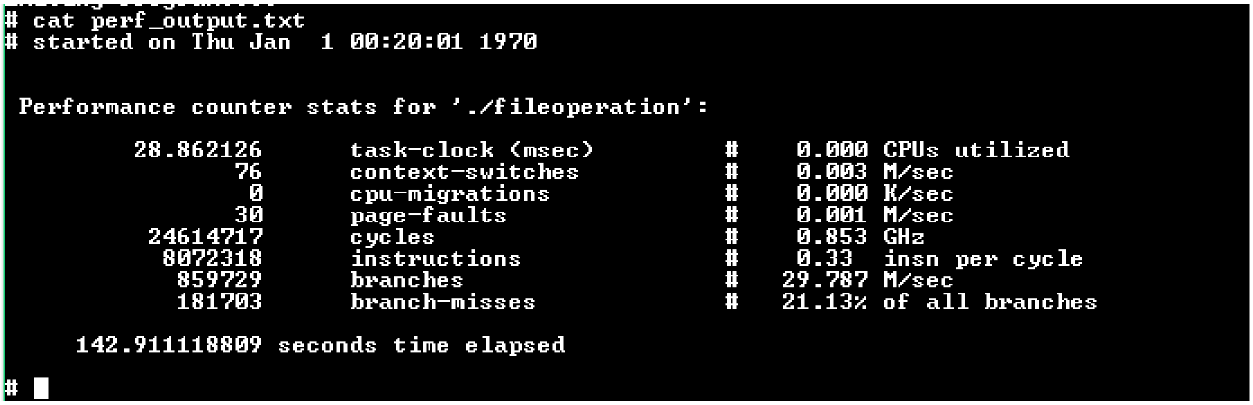
```
Exiting Program....
# cat ltrace_output.txt
  0.000000 SYS_332@SYS(-100, 0xb6fe0a00, 0xbee5aaf8, 4096) = 22 <0.012507>
  0.002575 mmap2@SYS(0, 4096, 3, 34)             = 0xb6fef000 <0.000814>
  0.001755 open@SYS("/lib//libc.so.0", 0, 00)    = 3 <0.001883>
  0.002452 fstat@SYS(3, 0xbee5b1ac)              = 0 <0.000619>
  0.001372 mmap2@SYS(0, 4096, 3, 34)             = 0xb6fee000 <0.000748>
  0.001502 read@SYS(3, "\177ELF\001\001\001", 4096) = 4096 <0.000599>
  0.001903 mmap2@SYS(0, 0x92000, 0, 34)          = 0xb6f49000 <0.000783>
  0.001491 mmap2@SYS(0xb6f49000, 0x6a6f4, 5, 18) = 0xb6f49000 <0.000842>
  0.001374 mmap2@SYS(0xb6fc3000, 4852, 3, 18)    = 0xb6fc3000 <0.000796>
  0.001471 mmap2@SYS(0xb6fc5000, 0x15828, 3, 50) = 0xb6fc5000 <0.000800>
  0.001357 close@SYS(3)                          = 0 <0.000544>
  0.001697 munmap@SYS(0xb6fee000, 4096)          = 0 <0.000715>
  0.001412 stat@SYS("/lib/ld-uClibc.so.0", 0xbee5ba90) = 0 <0.002232>
  0.002765 mmap2@SYS(0, 4096, 3, 34)             = 0xb6fee000 <0.000766>
  0.001588 ARCH_set_tls@SYS(0xb6fee490, 0xb6fee000, 0xb6feeb38, 0xb6fee490) = 0 <0.000831>
  0.001966 mprotect@SYS(0x23000, 4096, 1)        = 0 <0.000821>
  0.001483 mprotect@SYS(0xb6fc3000, 4096, 1)     = 0 <0.000721>
```

perf stat Output:



```
# perf stat -o perf_output.txt ./fileoperation
The following program demonstrates various file operation

Select operation number to proceed:
1. Print to standard out an interesting string using printf
2. Create a file
3. Operations Menu
4. Exit Program
1
Enter string to be printed on the std out using printf : hello
The entered string is : hello

Select operation number to proceed:
```

Cat perf_output.txt



```
# cat perf_output.txt
# started on Thu Jan  1 00:20:01 1970


 Performance counter stats for './fileoperation':

         28.862126      task-clock (msec)         #    0.000 CPUs utilized
                76      context-switches          #    0.003 M/sec
                 0      cpu-migrations            #    0.000 K/sec
                30      page-faults               #    0.001 M/sec
          24614717      cycles                    #    0.853 GHz
           8072318      instructions              #    0.33  insn per cycle
            859729      branches                  #   29.787 M/sec
            181703      branch-misses             #   21.13% of all branches

     142.911118809 seconds time elapsed

#
```

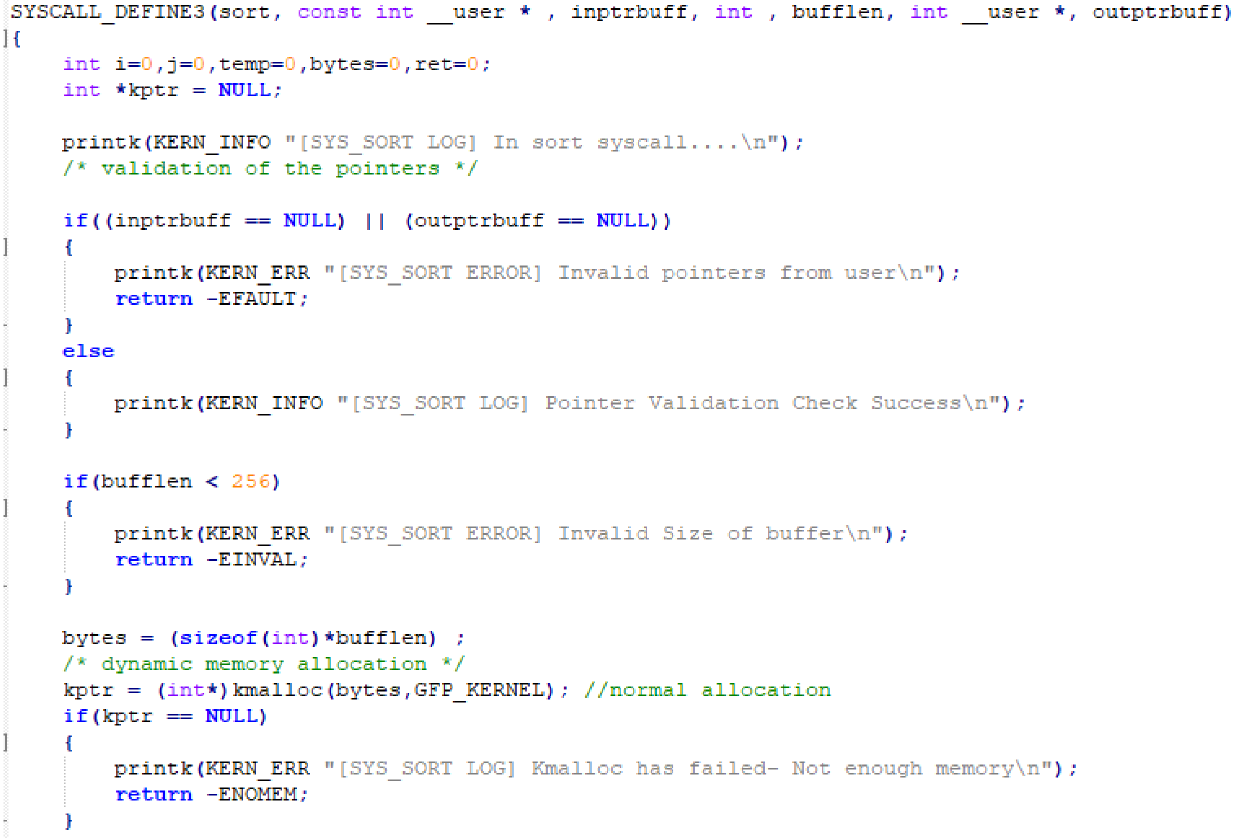
Q5. Implement Your Own System Call on BBG

You are to create your own system call that can sort an array of numbers in kernel mode.
The program takes a set of input parameters from user space including -Pointer to a buffer (input) ,Size of that buffer (entries or bytes) ,Pointer to a sorted buffer

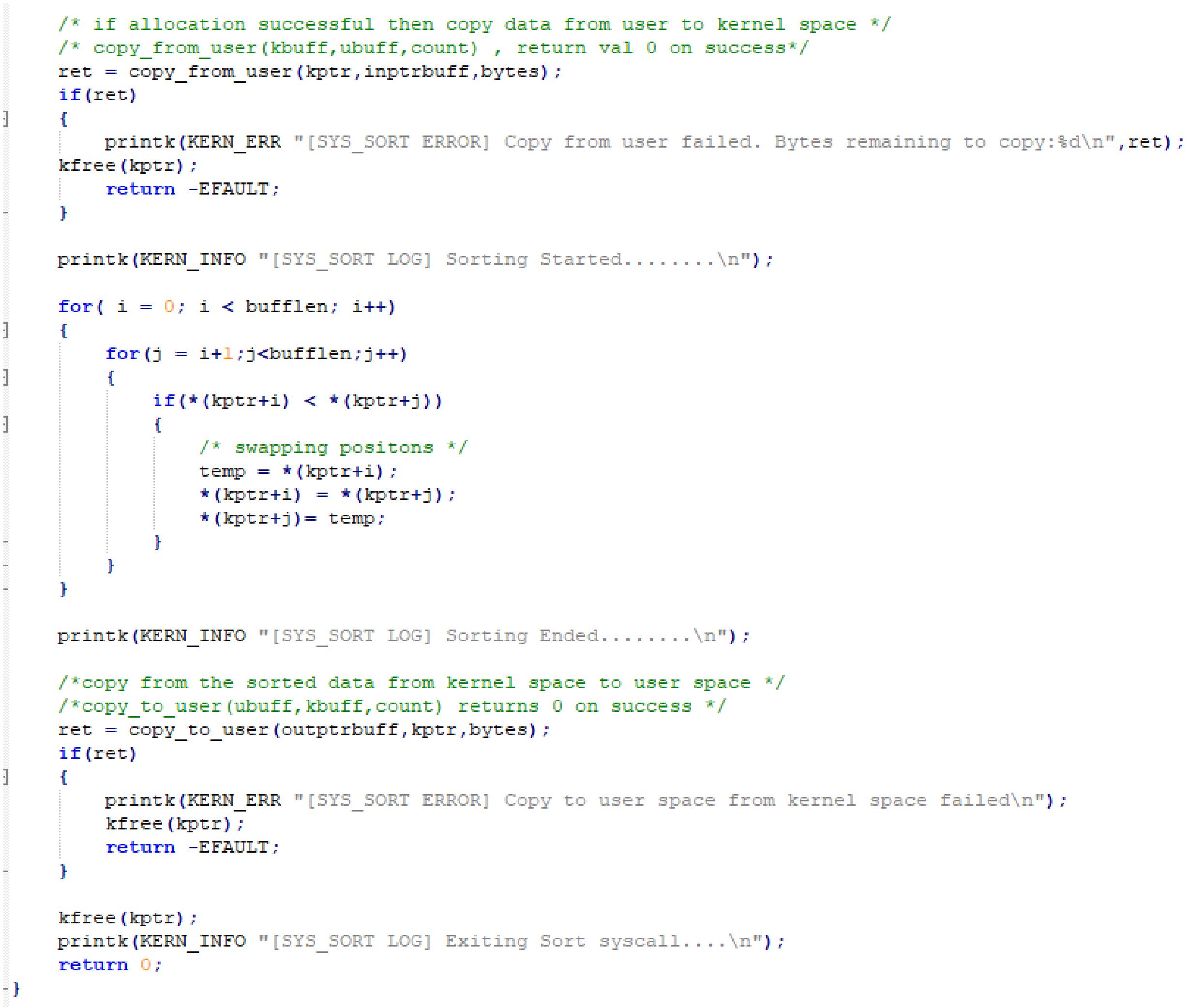
Ans: The steps for syscall are as follows:

1. Write the syscall starting with 'SYSCALL_DEFINE' in the 'sys.c' file. The path is → ~/buildroot/output/build/linux-<?>/kernel , open sys.c write the system call below the getpid syscall.

   The system call is as follows:

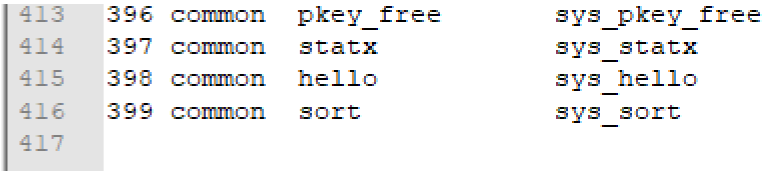


```
SYSCALL_DEFINE3(sort, const int __user * , inptrbuff, int , bufflen, int __user *, outptrbuff)
{
    int i=0,j=0,temp=0,bytes=0,ret=0;
    int *kptr = NULL;

    printk(KERN_INFO "[SYS_SORT LOG] In sort syscall....\n");
    /* validation of the pointers */

    if((inptrbuff == NULL) || (outptrbuff == NULL))
    {
        printk(KERN_ERR "[SYS_SORT ERROR] Invalid pointers from user\n");
        return -EFAULT;
    }
    else
    {
        printk(KERN_INFO "[SYS_SORT LOG] Pointer Validation Check Success\n");
    }

    if(bufflen < 256)
    {
        printk(KERN_ERR "[SYS_SORT ERROR] Invalid Size of buffer\n");
        return -EINVAL;
    }

    bytes = (sizeof(int)*bufflen) ;
    /* dynamic memory allocation */
    kptr = (int*)kmalloc(bytes,GFP_KERNEL); //normal allocation
    if(kptr == NULL)
    {
        printk(KERN_ERR "[SYS_SORT LOG] Kmalloc has failed- Not enough memory\n");
        return -ENOMEM;
    }
```



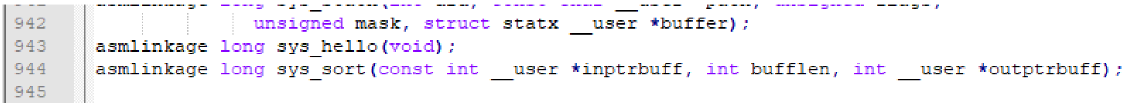
```
    /* if allocation successful then copy data from user to kernel space */
    /* copy_from_user(kbuff,ubuff,count) , return val 0 on success*/
    ret = copy_from_user(kptr,inptrbuff,bytes);
    if(ret)
    {
        printk(KERN_ERR "[SYS_SORT ERROR] Copy from user failed. Bytes remaining to copy:%d\n",ret);
    kfree(kptr);
        return -EFAULT;
    }

    printk(KERN_INFO "[SYS_SORT LOG] Sorting Started........\n");

    for( i = 0; i < bufflen; i++)
    {
        for(j = i+1;j<bufflen;j++)
        {
            if(*(kptr+i) < *(kptr+j))
            {
                /* swapping positons */
                temp = *(kptr+i);
                *(kptr+i) = *(kptr+j);
                *(kptr+j)= temp;
            }
        }
    }

    printk(KERN_INFO "[SYS_SORT LOG] Sorting Ended........\n");

    /*copy from the sorted data from kernel space to user space */
    /*copy_to_user(ubuff,kbuff,count) returns 0 on success */
    ret = copy_to_user(outptrbuff,kptr,bytes);
    if(ret)
    {
        printk(KERN_ERR "[SYS_SORT ERROR] Copy to user space from kernel space failed\n");
        kfree(kptr);
        return -EFAULT;
    }

    kfree(kptr);
    printk(KERN_INFO "[SYS_SORT LOG] Exiting Sort syscall....\n");
    return 0;
}
```

2. Make an entry in the syscall.tbl in the path-→ ~/buildroot/output/build/linux-<?>/arch/arm/tools at the end of the table along with the serial number.



```
396 common  pkey_free       sys_pkey_free
397 common  statx           sys_statx
398 common  hello           sys_hello
399 common  sort            sys_sort
```

3. Write the declaration of the syscall function in the syscall.h file in the path -> ~/buildroot/output/build/linux-<?>/include/linux

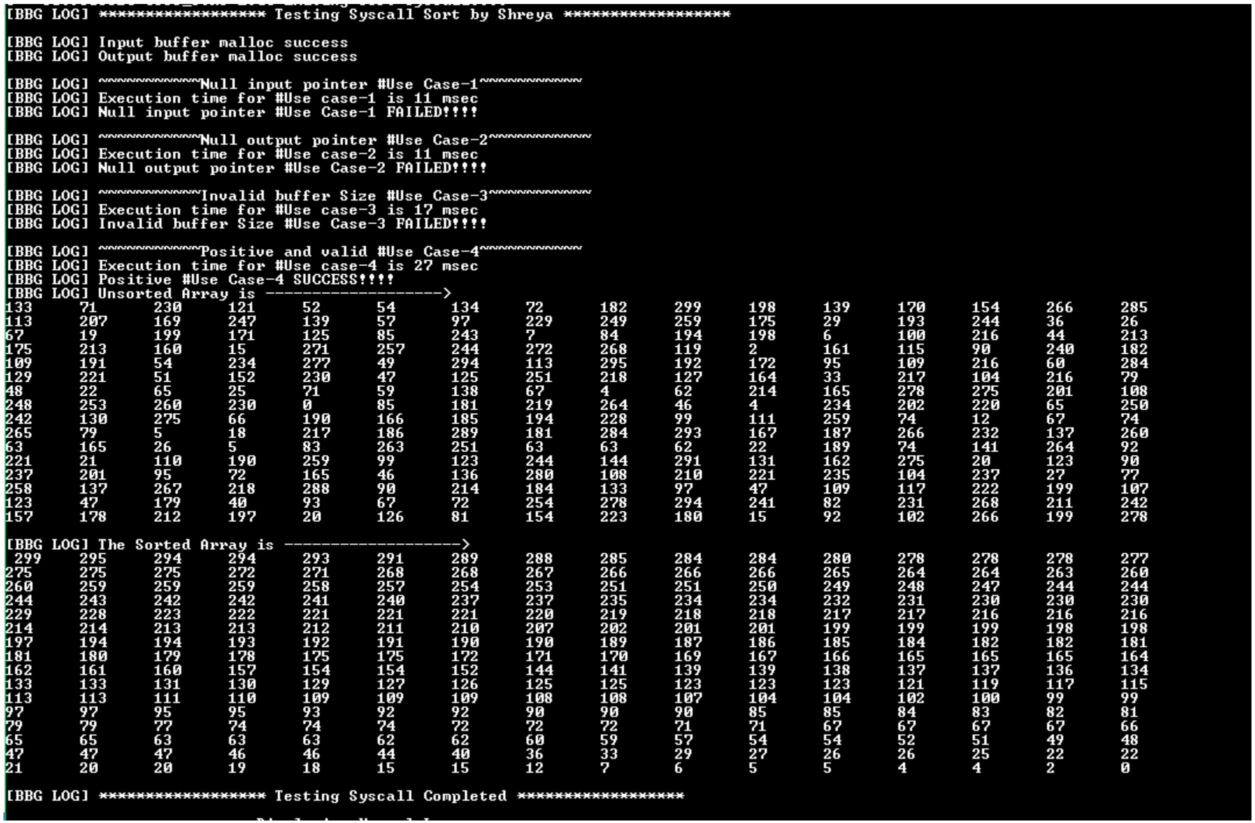


```
                unsigned mask, struct statx __user *buffer);
asmlinkage long sys_hello(void);
asmlinkage long sys_sort(const int __user *inptrbuff, int bufflen, int __user *outptrbuff);
```

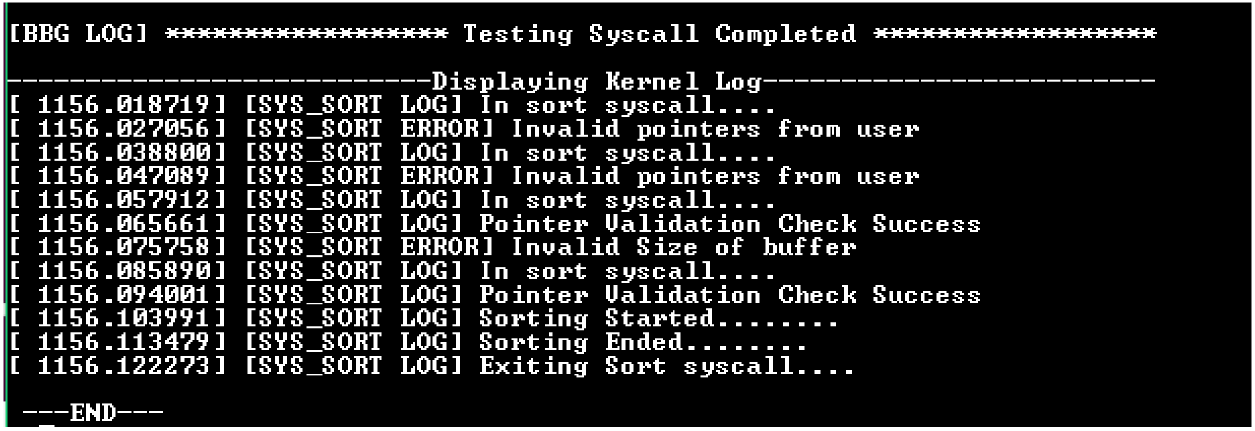
4. All the above mentioned folders are in Q5/BubbleSort of HW2 in a subfolder called Sysfile.
5. Create the test function to test the system call and cross compile it for BBG. Add the executable to the~/ buildroot/board/Beaglebone/root-overlay/usr/bin folder. In my repository the test function is in a subfolder of Q5 called Testfile.
6. Do sudo make linux-menuconfig and save the config as the default name and exit.

7. Do sudo make
8. Copy the sdcard.img in the sdcard and connect the FTDI cable and boot up the BBG.

Proof of execution on BBG using tera term

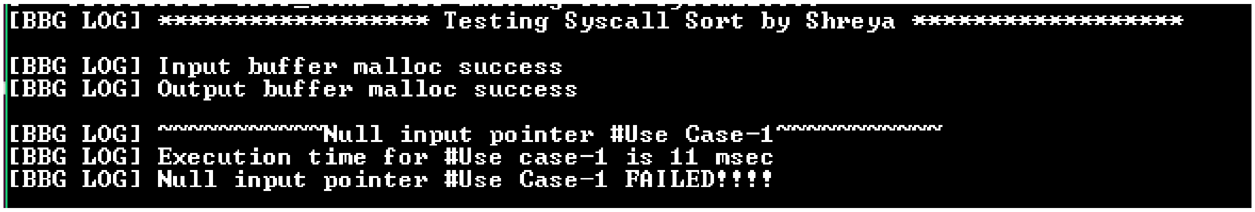


```
[BBG LOG] ****************** Testing Syscall Sort by Shreya ******************

[BBG LOG] Input buffer malloc success
[BBG LOG] Output buffer malloc success

[BBG LOG] ~~~~~~~~~~~~Null input pointer #Use Case-1~~~~~~~~~~~~
[BBG LOG] Execution time for #Use case-1 is 11 msec
[BBG LOG] Null input pointer #Use Case-1 FAILED!!!!

[BBG LOG] ~~~~~~~~~~~~Null output pointer #Use Case-2~~~~~~~~~~~~
[BBG LOG] Execution time for #Use case-2 is 11 msec
[BBG LOG] Null output pointer #Use Case-2 FAILED!!!!

[BBG LOG] ~~~~~~~~~~~~Invalid buffer Size #Use Case-3~~~~~~~~~~~~
[BBG LOG] Execution time for #Use case-3 is 17 msec
[BBG LOG] Invalid buffer Size #Use Case-3 FAILED!!!!

[BBG LOG] ~~~~~~~~~~~~Positive and valid #Use Case-4~~~~~~~~~~~~
[BBG LOG] Execution time for #Use case-4 is 27 msec
[BBG LOG] Positive #Use Case-4 SUCCESS!!!!
[BBG LOG] Unsorted Array is ------------------->
133     71      230     121     52      54      134     72      182     299     198     139     170     154     266     285
113     207     169     247     139     57      97      229     249     259     175     29      193     244     36      26
67      19      199     171     125     85      243     7       84      194     198     6       100     216     44      213
175     213     160     15      271     257     244     272     268     119     2       161     115     90      240     182
109     191     54      234     277     49      294     113     295     192     172     95      109     216     60      284
129     221     51      152     230     47      125     251     218     127     164     33      217     104     216     79
48      22      65      25      71      59      138     67      4       62      214     165     278     275     201     108
248     253     260     230     0       85      181     219     264     46      4       234     202     220     65      250
242     130     275     66      190     166     185     194     228     99      111     259     74      12      67      74
265     79      5       18      217     186     289     181     284     293     167     187     266     232     137     260
63      165     26      5       83      263     251     63      63      62      22      189     74      141     264     92
221     21      110     190     259     99      123     244     144     291     131     162     275     20      123     90
237     201     95      72      165     46      136     280     108     210     221     235     104     237     27      77
258     137     267     218     288     90      214     184     133     97      47      109     117     222     199     107
123     47      179     40      93      67      72      254     278     294     241     82      231     268     211     242
157     178     212     197     20      126     81      154     223     180     15      92      102     266     199     278

[BBG LOG] The Sorted Array is ------------------->
 299    295     294     294     293     291     289     288     285     284     284     280     278     278     278     277
275     275     275     272     271     268     268     267     266     266     266     265     264     264     263     260
260     259     259     259     258     257     254     253     251     251     250     249     248     247     244     244
244     243     242     242     241     240     237     237     235     234     234     232     231     230     230     230
229     228     223     222     221     221     221     220     219     218     218     217     217     216     216     216
214     214     213     213     212     211     210     207     202     201     201     199     199     199     198     198
197     194     194     193     192     191     190     190     189     187     186     185     184     182     182     181
181     180     179     178     175     175     172     171     170     169     167     166     165     165     165     164
162     161     160     157     154     154     152     144     141     139     139     138     137     137     136     134
133     133     131     130     129     127     126     125     125     123     123     123     121     119     117     115
113     113     111     110     109     109     109     108     108     107     104     104     102     100     99      99
97      97      95      95      93      92      92      90      90      90      85      85      84      83      82      81
79      79      77      74      74      74      72      72      72      71      71      67      67      67      67      66
65      65      63      63      63      62      62      60      59      57      54      54      52      51      49      48
47      47      47      46      46      44      40      36      33      29      27      26      26      25      22      22
21      20      20      19      18      15      15      12      7       6       5       5       4       4       2       0

[BBG LOG] ****************** Testing Syscall Completed ******************
```

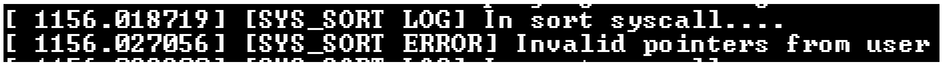


```
[BBG LOG] ******************* Testing Syscall Completed *******************

---------------------------Displaying Kernel Log--------------------------
[ 1156.018719] [SYS_SORT LOG] In sort syscall....
[ 1156.027056] [SYS_SORT ERROR] Invalid pointers from user
[ 1156.038800] [SYS_SORT LOG] In sort syscall....
[ 1156.047089] [SYS_SORT ERROR] Invalid pointers from user
[ 1156.057912] [SYS_SORT LOG] In sort syscall....
[ 1156.065661] [SYS_SORT LOG] Pointer Validation Check Success
[ 1156.075758] [SYS_SORT ERROR] Invalid Size of buffer
[ 1156.085890] [SYS_SORT LOG] In sort syscall....
[ 1156.094001] [SYS_SORT LOG] Pointer Validation Check Success
[ 1156.103991] [SYS_SORT LOG] Sorting Started........
[ 1156.113479] [SYS_SORT LOG] Sorting Ended........
[ 1156.122273] [SYS_SORT LOG] Exiting Sort syscall....

 ---END---
```

Use case 1 – invalid input buffer

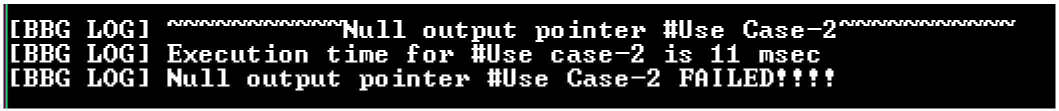


```
[BBG LOG] ****************** Testing Syscall Sort by Shreya ******************

[BBG LOG] Input buffer malloc success
[BBG LOG] Output buffer malloc success

[BBG LOG] ~~~~~~~~~~~~Null input pointer #Use Case-1~~~~~~~~~~~~
[BBG LOG] Execution time for #Use case-1 is 11 msec
[BBG LOG] Null input pointer #Use Case-1 FAILED!!!!
```

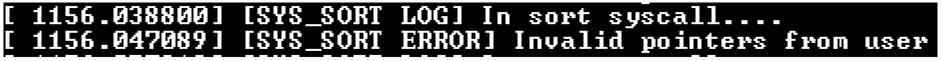


```
[ 1156.018719] [SYS_SORT LOG] In sort syscall....
[ 1156.027056] [SYS_SORT ERROR] Invalid pointers from user
```

Use case 2 – invalid output buffer

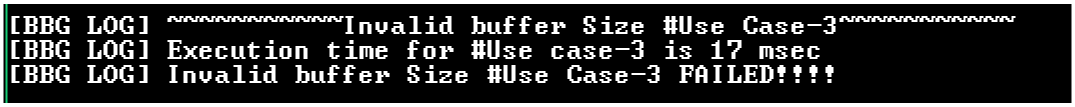


```
[BBG LOG] ~~~~~~~~~~~~~Null output pointer #Use Case-2~~~~~~~~~~~~~
[BBG LOG] Execution time for #Use case-2 is 11 msec
[BBG LOG] Null output pointer #Use Case-2 FAILED!!!!
```

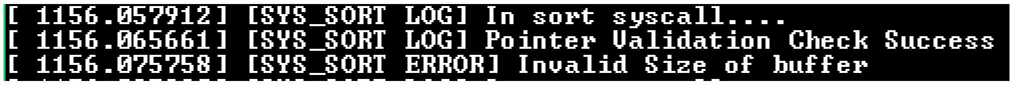


```
[ 1156.038800] [SYS_SORT LOG] In sort syscall....
[ 1156.047089] [SYS_SORT ERROR] Invalid pointers from user
```

Use case 3 – invalid buffer size

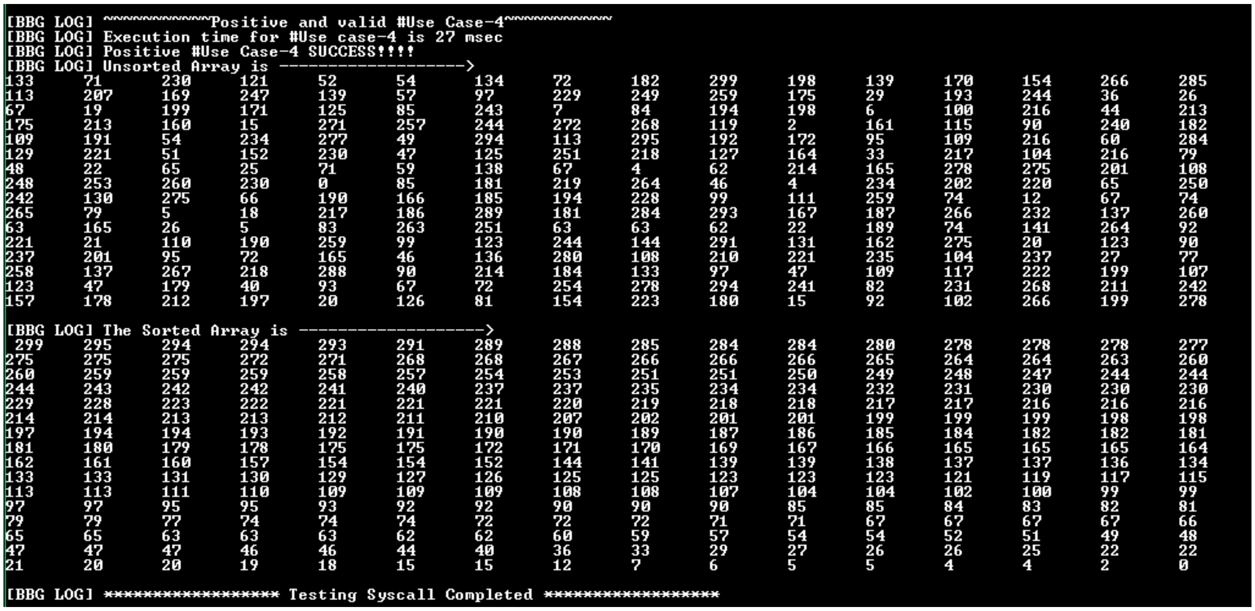


```
[BBG LOG] ~~~~~~~~~~~~~Invalid buffer Size #Use Case-3~~~~~~~~~~~~~
[BBG LOG] Execution time for #Use case-3 is 17 msec
[BBG LOG] Invalid buffer Size #Use Case-3 FAILED!!!!
```

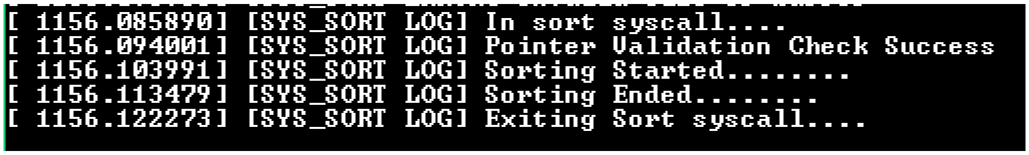


```
[ 1156.057912] [SYS_SORT LOG] In sort syscall....
[ 1156.065661] [SYS_SORT LOG] Pointer Validation Check Success
[ 1156.075758] [SYS_SORT ERROR] Invalid Size of buffer
```

Use case 4 – positive and valid case



```
[BBG LOG] ~~~~~~~~~~~Positive and valid #Use Case-4~~~~~~~~~~~
[BBG LOG] Execution time for #Use case-4 is 27 msec
[BBG LOG] Positive #Use Case-4 SUCCESS!!!!
[BBG LOG] Unsorted Array is ------------------->
133     71      230     121     52      54      134     72      182     299     198     139     170     154     266     285
113     207     169     247     139     57      97      229     249     259     175     29      193     244     36      26
67      19      199     171     125     85      243     7       84      194     198     6       100     216     44      213
175     213     160     15      271     257     244     272     268     119     2       161     115     90      240     182
109     191     54      234     277     49      294     113     295     192     172     95      109     216     60      284
129     221     51      152     230     47      125     251     218     127     164     33      217     104     216     79
48      22      65      25      71      59      138     67      4       62      214     165     278     275     201     108
248     253     260     230     0       85      181     219     264     46      4       234     202     220     65      250
242     130     275     66      190     166     185     194     228     99      111     259     74      12      67      74
265     79      5       18      217     186     289     181     284     293     167     187     266     232     137     260
63      165     26      5       83      263     251     63      63      62      22      189     74      141     264     92
221     21      110     190     259     99      123     244     144     291     131     162     275     20      123     90
237     201     95      72      165     46      136     280     108     210     221     235     104     237     27      77
258     137     267     218     288     90      214     184     133     97      47      109     117     222     199     107
123     47      179     40      93      67      72      254     278     294     241     82      231     268     211     242
157     178     212     197     20      126     81      154     223     180     15      92      102     266     199     278

[BBG LOG] The Sorted Array is ------------------->
 299    295     294     294     293     291     289     288     285     284     284     280     278     278     278     277
275     275     275     272     271     268     268     267     266     266     266     265     264     264     263     260
260     259     259     259     258     257     254     253     251     251     250     249     248     247     244     244
244     243     242     242     241     240     237     237     235     234     234     232     231     230     230     230
229     228     223     222     221     221     221     220     219     218     218     217     217     216     216     216
214     214     213     213     212     211     210     207     202     201     201     199     199     199     198     198
197     194     194     193     192     191     190     190     189     187     186     185     184     182     182     181
181     180     179     178     175     175     172     171     170     169     167     166     165     165     165     164
162     161     160     157     154     154     152     144     141     139     139     138     137     137     136     134
133     133     131     130     129     127     126     125     125     123     123     123     121     119     117     115
113     113     111     110     109     109     109     108     108     107     104     104     102     100     99      99
97      97      95      95      93      92      92      90      90      90      85      85      84      83      82      81
79      79      77      74      74      74      72      72      72      71      71      67      67      67      67      66
65      65      63      63      63      62      62      60      59      57      54      54      52      51      49      48
47      47      47      46      46      44      40      36      33      29      27      26      26      25      22      22
21      20      20      19      18      15      15      12      7       6       5       5       4       4       2       0

[BBG LOG] ****************** Testing Syscall Completed ******************
```

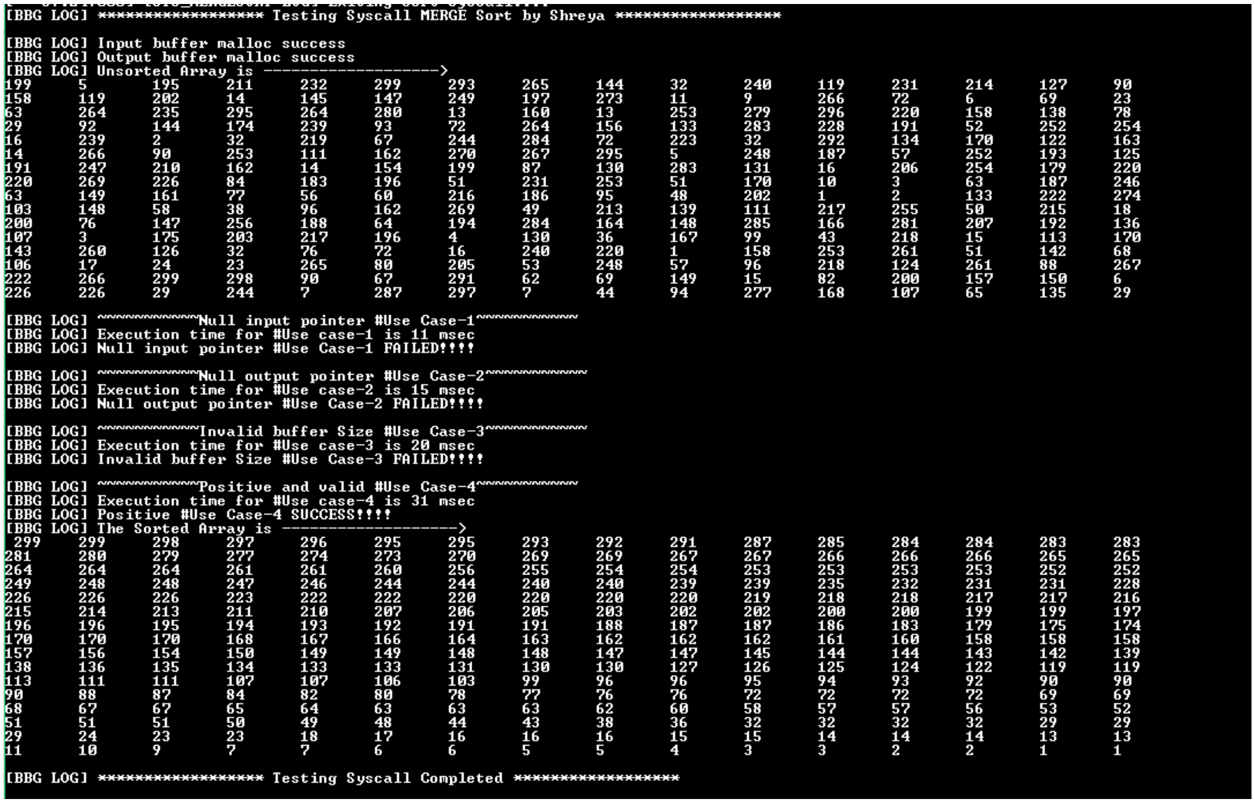


```
[ 1156.085890] [SYS_SORT LOG] In sort syscall....
[ 1156.094001] [SYS_SORT LOG] Pointer Validation Check Success
[ 1156.103991] [SYS_SORT LOG] Sorting Started........
[ 1156.113479] [SYS_SORT LOG] Sorting Ended........
[ 1156.122273] [SYS_SORT LOG] Exiting Sort syscall....
```

MERGE SORT(BONUS)

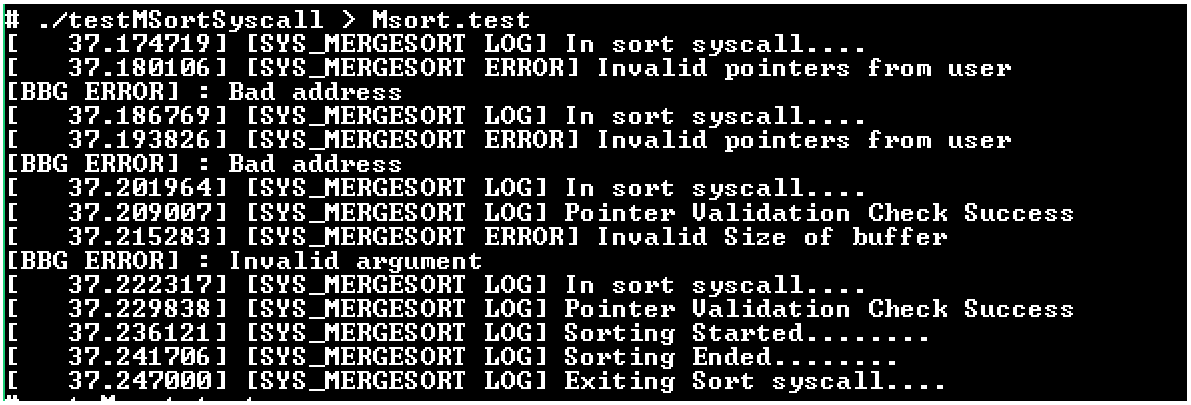
Sys files are available in the MergeSort file of Q5

Proof of execution:



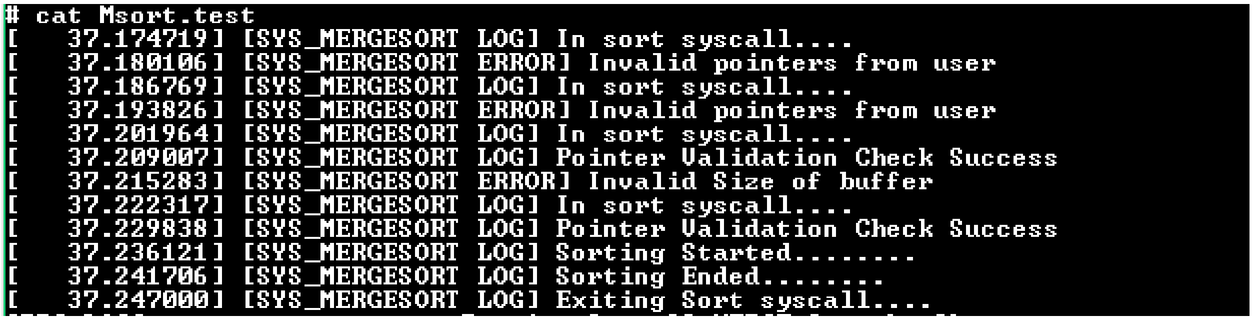
```
[BBG LOG] ******************* Testing Syscall MERGE Sort by Shreya *******************

[BBG LOG] Input buffer malloc success
[BBG LOG] Output buffer malloc success
[BBG LOG] Unsorted Array is -------------------->
199     5       195     211     232     299     293     265     144     32      240     119     231     214     127     90
158     119     202     14      145     147     249     197     273     11      9       266     72      6       69      23
63      264     235     295     264     280     13      160     13      253     279     296     220     158     138     78
29      92      144     174     239     93      72      264     156     133     283     228     191     52      252     254
16      239     2       32      219     67      244     284     72      223     32      292     134     170     122     163
14      266     90      253     111     162     270     267     295     5       248     187     57      252     193     125
191     247     210     162     14      154     199     87      130     283     131     16      206     254     179     220
220     269     226     84      183     196     51      231     253     51      170     10      3       63      187     246
63      149     161     77      56      60      216     186     95      48      202     1       2       133     222     274
103     148     58      38      96      162     269     49      213     139     111     217     255     50      215     18
200     76      147     256     188     64      194     284     164     148     285     166     281     207     192     136
107     3       175     203     217     196     4       130     36      167     99      43      218     15      113     170
143     260     126     32      76      72      16      240     220     1       158     253     261     51      142     68
106     17      24      23      265     80      205     53      248     57      96      218     124     261     88      267
222     266     299     298     90      67      291     62      69      149     15      82      200     157     150     6
226     226     29      244     7       287     297     7       44      94      277     168     107     65      135     29

[BBG LOG] ~~~~~~~~~~~~~Null input pointer #Use Case-1~~~~~~~~~~~~~
[BBG LOG] Execution time for #Use case-1 is 11 msec
[BBG LOG] Null input pointer #Use Case-1 FAILED!!!!

[BBG LOG] ~~~~~~~~~~~~~Null output pointer #Use Case-2~~~~~~~~~~~~~
[BBG LOG] Execution time for #Use case-2 is 15 msec
[BBG LOG] Null output pointer #Use Case-2 FAILED!!!!

[BBG LOG] ~~~~~~~~~~~~~Invalid buffer Size #Use Case-3~~~~~~~~~~~~~
[BBG LOG] Execution time for #Use case-3 is 20 msec
[BBG LOG] Invalid buffer Size #Use Case-3 FAILED!!!!

[BBG LOG] ~~~~~~~~~~~~~Positive and valid #Use Case-4~~~~~~~~~~~~~
[BBG LOG] Execution time for #Use case-4 is 31 msec
[BBG LOG] Positive #Use Case-4 SUCCESS!!!!
[BBG LOG] The Sorted Array is -------------------->
 299    299     298     297     296     295     295     293     292     291     287     285     284     284     283     283
281     280     279     277     274     273     270     269     269     267     267     266     266     266     265     265
264     264     264     261     261     260     256     255     254     254     253     253     253     253     252     252
249     248     248     247     246     244     244     240     240     239     239     235     232     231     231     228
226     226     226     223     222     222     220     220     220     220     219     218     218     217     217     216
215     214     213     211     210     207     206     205     203     202     202     200     200     199     199     197
196     196     195     194     193     192     191     191     188     187     187     186     183     179     175     174
170     170     170     168     167     166     164     163     162     162     162     161     160     158     158     158
157     156     154     150     149     149     148     148     147     147     145     144     144     143     142     139
138     136     135     134     133     133     131     130     130     127     126     125     124     122     119     119
113     111     111     107     107     106     103     99      96      96      95      94      93      92      90      90
90      88      87      84      82      80      78      77      76      76      72      72      72      72      69      69
68      67      67      65      64      63      63      63      62      60      58      57      57      56      53      52
51      51      51      50      49      48      44      43      38      36      32      32      32      32      29      29
29      24      23      23      18      17      16      16      16      15      15      14      14      14      13      13
11      10      9       7       7       6       6       5       5       4       3       3       2       2       1       1

[BBG LOG] ******************* Testing Syscall Completed *******************
```

Error values of use cases:



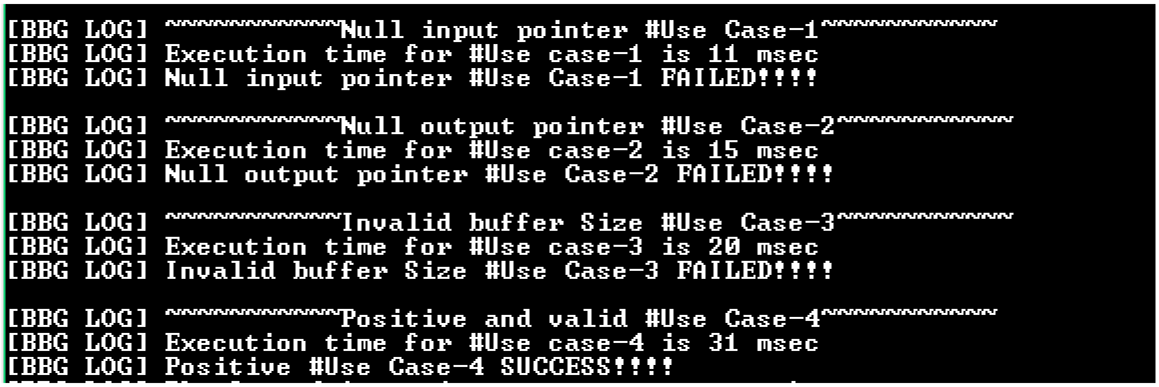
```
# ./testMSortSyscall > Msort.test
[   37.174719] [SYS_MERGESORT LOG] In sort syscall....
[   37.180106] [SYS_MERGESORT ERROR] Invalid pointers from user
[BBG ERROR] : Bad address
[   37.186769] [SYS_MERGESORT LOG] In sort syscall....
[   37.193826] [SYS_MERGESORT ERROR] Invalid pointers from user
[BBG ERROR] : Bad address
[   37.201964] [SYS_MERGESORT LOG] In sort syscall....
[   37.209007] [SYS_MERGESORT LOG] Pointer Validation Check Success
[   37.215283] [SYS_MERGESORT ERROR] Invalid Size of buffer
[BBG ERROR] : Invalid argument
[   37.222317] [SYS_MERGESORT LOG] In sort syscall....
[   37.229838] [SYS_MERGESORT LOG] Pointer Validation Check Success
[   37.236121] [SYS_MERGESORT LOG] Sorting Started........
[   37.241706] [SYS_MERGESORT LOG] Sorting Ended........
[   37.247000] [SYS_MERGESORT LOG] Exiting Sort syscall....
```

Kernel log using dmesg:



```
# cat Msort.test
[   37.174719] [SYS_MERGESORT LOG] In sort syscall....
[   37.180106] [SYS_MERGESORT ERROR] Invalid pointers from user
[   37.186769] [SYS_MERGESORT LOG] In sort syscall....
[   37.193826] [SYS_MERGESORT ERROR] Invalid pointers from user
[   37.201964] [SYS_MERGESORT LOG] In sort syscall....
[   37.209007] [SYS_MERGESORT LOG] Pointer Validation Check Success
[   37.215283] [SYS_MERGESORT ERROR] Invalid Size of buffer
[   37.222317] [SYS_MERGESORT LOG] In sort syscall....
[   37.229838] [SYS_MERGESORT LOG] Pointer Validation Check Success
[   37.236121] [SYS_MERGESORT LOG] Sorting Started........
[   37.241706] [SYS_MERGESORT LOG] Sorting Ended........
[   37.247000] [SYS_MERGESORT LOG] Exiting Sort syscall....
```

Use cases:



```
[BBG LOG] ~~~~~~~~~~~Null input pointer #Use Case-1~~~~~~~~~~~
[BBG LOG] Execution time for #Use case-1 is 11 msec
[BBG LOG] Null input pointer #Use Case-1 FAILED!!!!

[BBG LOG] ~~~~~~~~~~~Null output pointer #Use Case-2~~~~~~~~~~~
[BBG LOG] Execution time for #Use case-2 is 15 msec
[BBG LOG] Null output pointer #Use Case-2 FAILED!!!!

[BBG LOG] ~~~~~~~~~~~Invalid buffer Size #Use Case-3~~~~~~~~~~~
[BBG LOG] Execution time for #Use case-3 is 20 msec
[BBG LOG] Invalid buffer Size #Use Case-3 FAILED!!!!

[BBG LOG] ~~~~~~~~~~~Positive and valid #Use Case-4~~~~~~~~~~~
[BBG LOG] Execution time for #Use case-4 is 31 msec
[BBG LOG] Positive #Use Case-4 SUCCESS!!!!
```

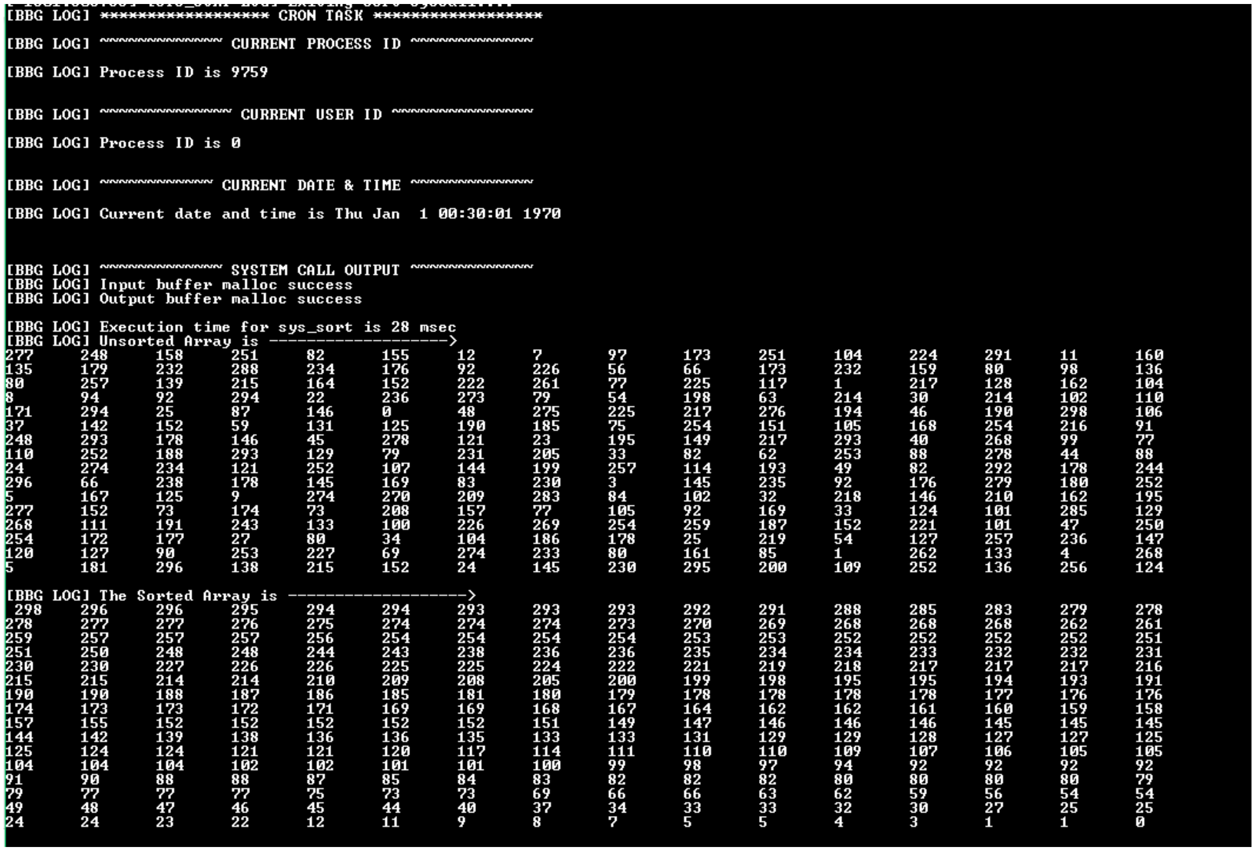
Q6. Create a CRON/Systemd task on BBG

Steps for Cron:

1. In menuconfig->Target Packages-> System tools -> select dcron
2. Sudo make linux-menuconfig and save the config as default.
3. Sudo make, burn the sd card image and boot up the BBG.
4. mkdir -p /var/spool/cron/crontabs
5. Open crontab using crontab -e
6. To run the crontask every 10 mins- Insert,type the command : */10 * * * * /usr/bin/crontask >> crontest.log and :wq to quit

The files pertaining to cron task are in Q6 folder of HW repository along with the output log file called cronTest.log

Proof of Execution



```
[BBG LOG] ****************** CRON TASK ******************

[BBG LOG] ~~~~~~~~~~~~~~ CURRENT PROCESS ID ~~~~~~~~~~~~~~

[BBG LOG] Process ID is 9759


[BBG LOG] ~~~~~~~~~~~~~~~ CURRENT USER ID ~~~~~~~~~~~~~~~~

[BBG LOG] Process ID is 0


[BBG LOG] ~~~~~~~~~~~~~ CURRENT DATE & TIME ~~~~~~~~~~~~~~

[BBG LOG] Current date and time is Thu Jan  1 00:30:01 1970



[BBG LOG] ~~~~~~~~~~~~~~ SYSTEM CALL OUTPUT ~~~~~~~~~~~~~~
[BBG LOG] Input buffer malloc success
[BBG LOG] Output buffer malloc success

[BBG LOG] Execution time for sys_sort is 28 msec
[BBG LOG] Unsorted Array is -------------------->
277     248     158     251     82      155     12      7       97      173     251     104     224     291     11      160
135     179     232     288     234     176     92      226     56      66      173     232     159     80      98      136
80      257     139     215     164     152     222     261     77      225     117     1       217     128     162     104
8       94      92      294     22      236     273     79      54      198     63      214     30      214     102     110
171     294     25      87      146     0       48      275     225     217     276     194     46      190     298     106
37      142     152     59      131     125     190     185     75      254     151     105     168     254     216     91
248     293     178     146     45      278     121     23      195     149     217     293     40      268     99      77
110     252     188     293     129     79      231     205     33      82      62      253     88      278     44      88
24      274     234     121     252     107     144     199     257     114     193     49      82      292     178     244
296     66      238     178     145     169     83      230     3       145     235     92      176     279     180     252
5       167     125     9       274     270     209     283     84      102     32      218     146     210     162     195
277     152     73      174     73      208     157     77      105     92      169     33      124     101     285     129
268     111     191     243     133     100     226     269     254     259     187     152     221     101     47      250
254     172     177     27      80      34      104     186     178     25      219     54      127     257     236     147
120     127     90      253     227     69      274     233     80      161     85      1       262     133     4       268
5       181     296     138     215     152     24      145     230     295     200     109     252     136     256     124

[BBG LOG] The Sorted Array is -------------------->
 298    296     296     295     294     294     293     293     293     292     291     288     285     283     279     278
278     277     277     276     275     274     274     274     273     270     269     268     268     268     262     261
259     257     257     257     256     254     254     254     254     253     253     252     252     252     252     251
251     250     248     248     244     243     238     236     236     235     234     234     233     232     232     231
230     230     227     226     226     225     225     224     222     221     219     218     217     217     217     216
215     215     214     214     210     209     208     205     200     199     198     195     195     194     193     191
190     190     188     187     186     185     181     180     179     178     178     178     178     177     176     176
174     173     173     172     171     169     169     168     167     164     162     162     161     160     159     158
157     155     152     152     152     152     152     151     149     147     146     146     146     145     145     145
144     142     139     138     136     136     135     133     133     131     129     129     128     127     127     125
125     124     124     121     121     120     117     114     111     110     110     109     107     106     105     105
104     104     104     102     102     101     101     100     99      98      97      94      92      92      92      92
91      90      88      88      87      85      84      83      82      82      82      80      80      80      80      79
79      77      77      77      75      73      73      69      66      66      63      62      59      56      54      54
49      48      47      46      45      44      40      37      34      33      33      32      30      27      25      25
24      24      23      22      12      11      9       8       7       5       5       4       3       1       1       0
```

The cronlog is included in the Q6 of HW2 in my repository.

References:

1. http://man7.org/linux/man-pages/man3/getline.3.html
2. https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.3.0/com.ibm.zos.v2r3.bpxbd00/rtchm.htm
3. Makefile : https://stackoverflow.com/questions/1484817/how-do-i-make-a-simple-makefile-for-gcc-on-linux
4. file permissions : https://stackoverflow.com/questions/8812959/how-to-read-linux-file-permission-programmatically-in-c-c
5. https://www.geeksforgeeks.org/fgetc-fputc-c/
6. https://stackoverflow.com/questions/30428615/taking-user-input-and-storing-it-in-an-array-of-strings-in-c
7. https://www.includehelp.com/c-programs/find-size-of-file.aspx
8. https://www.geeksforgeeks.org/iterative-merge-sort/
9. Syscall lecture 3.