HW3 REPORT

1. Set up Ethernet Networking

Ans: The following are the steps to set up ethernet networking on BBG to enable scp from host.
Reference: https://bootlin.com/doc/training/buildroot/buildroot-labs.pdf
(Note it worked for me this way! May or may not work for others)

1. Connect the ethernet cable to the BBG and the FTD cable to the PC.
2. In the Virtual Box Manager go to setting -> network-> enable Adapter 2 to bridged adapter.
3. Do ifconfig on both BBG and host and check the configuration of 'etho' on BBG and 'enp0s3' in host. Note the inet address of enp0s3.
4. In ubuntu 18.04 go to wired connections , Wired settings and open Network box. Go to settings gear icon next to the on/off button for enp0s3 and go to IPv4. Click Manual and enter the same inet enp0s3 address below the address box, under netmask enter 255.255.255.0 and click apply. Then click off next to the gear icon and turn the ethernet on again. When you do ifconfig again you will see the same inet address for enp0s3. This way we configure network on the host statically.
   Ifconfig on Host

```
shreya1809@shreya1809:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::2881:8c8b:f78c:e1bb  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:7d:1f:52  txqueuelen 1000  (Ethernet)
        RX packets 36  bytes 3986 (3.9 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 388  bytes 42678 (42.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.0.38  netmask 255.255.255.0  broadcast 10.0.0.255
        inet6 fe80::143d:2575:31f:41a  prefixlen 64  scopeid 0x20<link>
        inet6 2601:281:8500:92d:d871:2148:c43:23fa  prefixlen 64  scopeid 0x0<global>
        inet6 2601:281:8500:92d:8394:75d5:9408:6037  prefixlen 64  scopeid 0x0<global>
        ether 08:00:27:bf:d0:b0  txqueuelen 1000  (Ethernet)
        RX packets 7219  bytes 6886311 (6.8 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1215  bytes 135348 (135.3 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 661  bytes 51001 (51.0 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 661  bytes 51001 (51.0 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

5. Next we enable packages in Menuconfig. The packages to enable are in Target packages -> Networking Applications and select all from dhcp (ISC) to dhcpdump. Select dropbear and all its options. Save config as default and sudo make.

6. Create a folder called felabs in buildroot/board. Create several folder in folder such that the path is felabs/BBG/rootfs-overlay/etc/network/ and inside network create a file called interfaces. The contents of the file should be
   auto lo
   iface lo inet loopback auto
   eth0
   iface eth0 inet static
   address 10.0.0.16
   netmask 255.255.255.0

I chose the ip address 10.0.0.16 to be configured by eth0 by default. Add this path to the file system overlay in menuconfig.

7. sudo make the changes and observer buildroot/output/target/ etc/network/interfaces to see if your changes have been reflected in this file.
8. Burn the sd card image into the sd card and boot uo the BBG. You will see several new packages added in the bootup messages.

```
Starting sslh: OK
Starting network: [    2.705769] net eth0: initializing cpsw version 1.12 (0)
[    2.713227] libphy: PHY 4a101000.mdio:00 not found
[    2.718238] net eth0: phy "4a101000.mdio:00" not found on slave 0, err -19
[    2.736617] IPv6: ADDRCONF(NETDEV_UP): eth0: link is not ready
OK
Starting dhcpcd...
no interfaces have a carrier
forked to background, child pid 114
Starting dropbear sshd: [    3.244369] random: dropbear: uninitialized urandom read (32 bytes read)
OK
[    3.384230] random: ssh-keygen: uninitialized urandom read (32 bytes read)
Starting sshd: OK
Starting DHCP server: FAIL
Starting cron ... done.
[    3.729883] urandom_read: 1 callbacks suppressed
[    3.729896] random: crond: uninitialized urandom read (6 bytes read)

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

eth0 on putty:

```
[    3.605746] random: sshd: uninitialized urandom read (32 bytes read)
[    3.875160] random: crond: uninitialized urandom read (6 bytes read)
[    4.890159] cpsw 4a100000.ethernet eth0: Link is Up - 100Mbps/Full - flow con
trol rx/tx
[    4.899252] IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
[    4.939122] random: dhcpcd: uninitialized urandom read (120 bytes read)
[   15.955165] random: dropbear: uninitialized urandom read (32 bytes read)
[   29.961644] random: dropbear: uninitialized urandom read (32 bytes read)
```

9. Open putty and enter the address 10.0.0.16 under ip address port 22, SSH save setting as BBG and open. Login as root enter password if applicable and do ifconfig
   Ifconfig on BBG:

```
# ifconfig
eth0      Link encap:Ethernet  HWaddr 38:D2:69:5D:4B:BB
          inet addr:10.0.0.16  Bcast:0.0.0.0  Mask:255.255.255.0
          inet6 addr: 2601:281:8500:92d::af81/128 Scope:Global
          inet6 addr: 2601:281:8500:92d:509:331e:d2b4:6768/64 Scope:Global
          inet6 addr: fe80::9967:8f0d:3027:d780/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2422 errors:0 dropped:0 overruns:0 frame:0
          TX packets:421 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:368708 (360.0 KiB)  TX bytes:53416 (52.1 KiB)
          Interrupt:46

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:6 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:504 (504.0 B)  TX bytes:504 (504.0 B)
```

Notice eth0 inet address is 10.0.0.16.
At this point ping 10.0.0.16 on the host to see if it works.

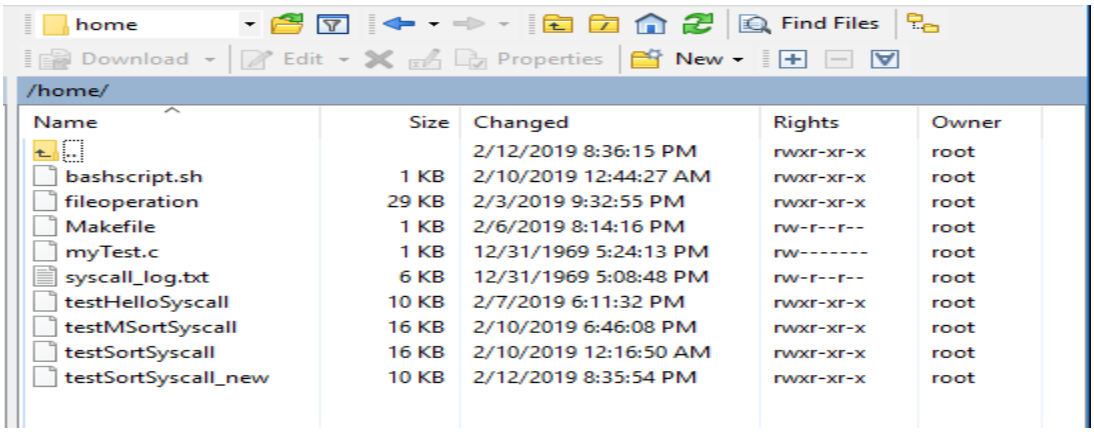
10. Try ssh root@10.0.0.16 and see if it works. Then scp and send the file you want to send.

```
shreya1809@shreya1809:~$ sudo scp myTest.c root@10.0.0.16:/home
The authenticity of host '10.0.0.16 (10.0.0.16)' can't be established.
ECDSA key fingerprint is SHA256:hUpNjuy+q3q+WkqyOn0q93ul/grha6UJS9XTlnjnJT0.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.0.16' (ECDSA) to the list of known hosts.
root@10.0.0.16's password:
myTest.c                                      100%   79     3.4KB/s   00:00
shreya1809@shreya1809:~$
```

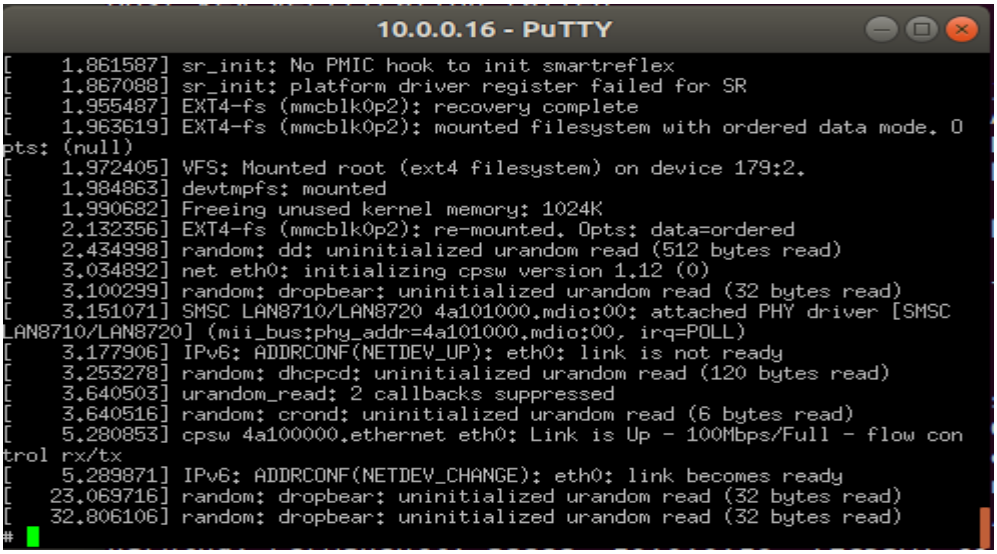
11. on putty via ssh

```
# cd ../../home
# ls
Makefile            myTest.c            testMSortSyscall
bashscript.sh       syscall_log.txt     testSortSyscall
fileoperation       testHelloSyscall    testSortSyscall_new
#
```

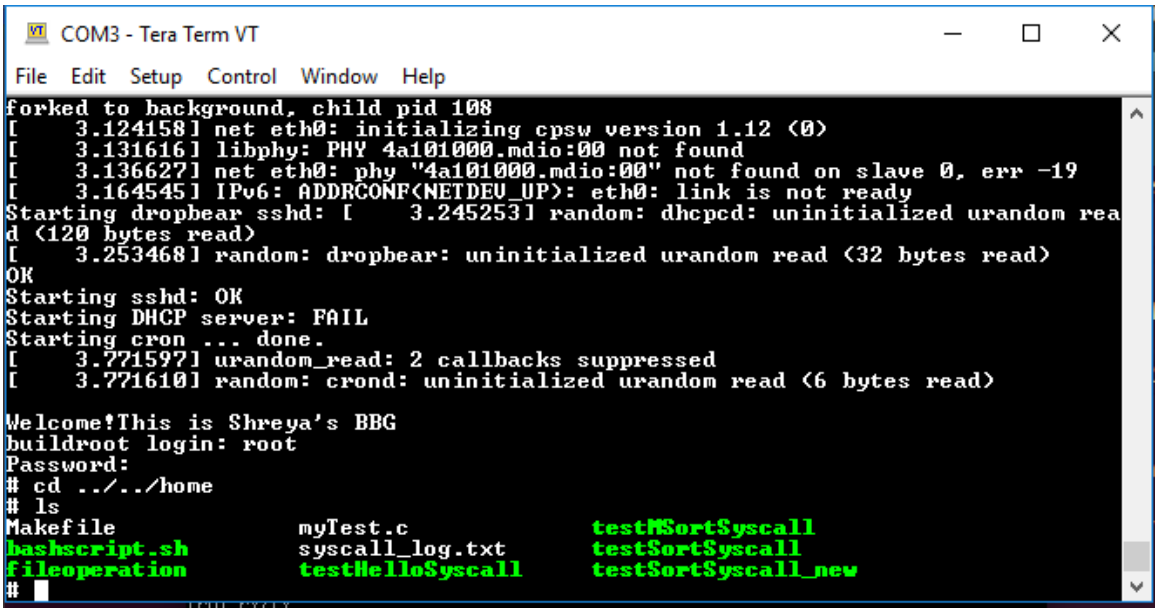
12. On WinScp



13. Dmesg on putty



14. On Tera term



15. In case the connect gets lost, on/off the wired connections, restart the BBG or restart the VM.

2. Remote debugging your application with GBD

Reference: https://www.thegeekstuff.com/2014/04/gdbserver-example/

Ans: The steps to be followed are:

1. Enable several packages in buildroot. The menuonfig configuration at this moment are:
    - Build Options -> enable compiler cache,use relative paths,strip target binaries
    - System Configuration -> Enable root login with password(optional),run a getty…, remount root file system .., purge unwanted locals
    - Kernel -> build a device tree DTB
    - Target packages -> debugging profiling and benchmark -> gdb (all options), ltrace, strace
    - Toolchain -> enable wchar support, thread library debugging,Compile and install ulibc utilities, gcc compiler version(gcc 8.x), enable c++ support, build cross gdb for host, tui support. Enable MMU support.
2. Sudo make and burn the image into the sdcard
3. Check if gdbserver is present by just typing 'gdbserver'
4. Start the gdbserver Using the command #gdbserver –multi localhost:2008

```
# gdbserver --multi localhost:2008
Listening on port 2008
```

   The above launches the gdb on the target without a program name waits for a debugger to connect to it on port 2008.
5. In the host now install gdb-multiarch and on the terminal: gdb-multiarch followed by program name eg $ gdb-multiarch testCProg.
6. Host commands connecting to the target : (gdb) target extended-remote 10.0.0.16:2008
7. Pushing out-of-tree executable to the target : (gdb) remote put testCProg testCProg
8. Select the file to debug. This should be set to the filename on the target system in our case both the target file name and the host file name is testCProg. (gdb) set remote exec-file testCProg
9. To set break point select the file:'(gdb) file testCProg' and load the symbols and then '(gdb) b main' to put a breakpoint at main.
10. To execute the program "(gdb) run" and type 'n' to single step it.

On BBG target:

```
# gdbserver --multi localhost:2008
Listening on port 2008
Remote debugging from host 10.0.0.38
Process /root/testCProg created; pid = 25432
Hello from shreya

Child exited with status 0
```

On host linux:

```
shreya1809@shreya1809:~$ gdb-multiarch testCProg
GNU gdb (Ubuntu 8.1-0ubuntu3) 8.1.0.20180409-git
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from testCProg...done.
(gdb) target extended-remote 10.0.0.16:2008
Remote debugging using 10.0.0.16:2008
(gdb) remote put testCProg testCProg
Successfully sent file "testCProg".
(gdb) set remote exec-file testCProg
(gdb) file testCProg
Load new symbol table from "testCProg"? (y or n) y
Reading symbols from testCProg...done.
(gdb) b main
Breakpoint 1 at 0x104c0: file myTest.c, line 6.
(gdb) run
Starting program: /home/shreya1809/testCProg
Reading /lib/ld-uClibc.so.0 from remote target...
```

```
Reading /lib/ld-uClibc.so.0 from remote target...
Reading /lib//libc.so.0 from remote target...

Breakpoint 1, main () at myTest.c:6
6               printf("Hello from shreya\n");
(gdb) finish
"finish" not meaningful in the outermost frame.
(gdb) n
7               return 0;
(gdb) n
8       }
(gdb) n
0xb6ee86f0 in __uClibc_main () from target:/lib//libc.so.0
(gdb) n
Single stepping until exit from function __uClibc_main,
which has no line number information.
[Inferior 1 (process 25432) exited normally]
(gdb)
```

Example of file operation code from HW2

On target BBG:

```
Listening on port 2008
Remote debugging from host 10.0.0.38
Process /root/fileoperation created; pid = 28046
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 : Remote GDB debugging ex
ample
The entered string is : Remote GDB debugging example

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
4
Exiting Program....

Child exited with status 0
```

On host linux:

```
shreya1809@shreya1809:~$ gdb-multiarch fileoperation
GNU gdb (Ubuntu 8.1-0ubuntu3) 8.1.0.20180409-git
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from fileoperation...done.
(gdb) target extended-remote 10.0.0.16:2008
Remote debugging using 10.0.0.16:2008
(gdb) remote put fileoperation fileoperation
Successfully sent file "fileoperation".
(gdb) set remote exec-file fileoperation
(gdb) file fileoperation
Load new symbol table from "fileoperation"? (y or n) y
Reading symbols from fileoperation...done.
(gdb) b main
Breakpoint 1 at 0x10980: file fileoperations.c, line 24.
(gdb) run
Starting program: /home/shreya1809/fileoperation
Reading /lib/ld-uClibc.so.0 from remote target...
```

```
Reading /lib/ld-uClibc.so.0 from remote target...
Reading /lib//libc.so.0 from remote target...

Breakpoint 1, main () at fileoperations.c:24
24          char *str = NULL;
(gdb) n
26          printf("The following program demonstrates various file operation\n");
(gdb) n
29              printf("\n");
(gdb) n
30              printf("Select operation number to proceed:\n");
(gdb) n
31              printf("1. Print to standard out an interesting string using printf\n");
(gdb) n
32              printf("2. Create a file\n");
(gdb) n
33              printf("3. Operations Menu\n");
(gdb) n
34              printf("4. Exit Program\n");
(gdb) n
35              c = getchar();
(gdb) n
36              x = getchar();
(gdb) n
37              if (c != '1' && c != '2' && c != '3' && c !='4')
(gdb) n
42              else if(c == '1')
(gdb) n
44                  printf("Enter string to be printed on the std out using printf : ");
(gdb) n
45                  str = readstringinput();
(gdb) n
                    printf("Enter string to be printed on the std out using printf : ");
(gdb) n
45                  str = readstringinput();
(gdb) n
46                  printf("The entered string is : %s\n",str);
(gdb) n
47                  free(str);
(gdb) n
48                  continue;
(gdb) n
29              printf("\n");
(gdb) n
30              printf("Select operation number to proceed:\n");
(gdb) n
31              printf("1. Print to standard out an interesting string using printf\n");
(gdb) n
32              printf("2. Create a file\n");
(gdb) n
33              printf("3. Operations Menu\n");
(gdb) n
34              printf("4. Exit Program\n");
(gdb) n
35              c = getchar();
(gdb) n
n
36              x = getchar();
(gdb) n
37              if (c != '1' && c != '2' && c != '3' && c !='4')
(gdb) n
42              else if(c == '1')
(gdb) n
50              else if (c == '2')
(gdb) n
50                  else if (c == '2')
(gdb) n
63                  else if( c == '4')
(gdb) n
65                      printf("Exiting Program....\n");
(gdb) n
66                      exit(0);
(gdb) n
[Inferior 1 (process 28046) exited normally]
(gdb)
```

GDBINIT

Contents of gdbinit file:

```
file fileoperation
target extended-remote 10.0.0.16:2008
remote put fileoperation fileoperation
set remote exec-file fileoperation
b main
run
~
~
~
~
~
~
~
```

Debugging using host initialization file gdbinit

```
shreya1809@shreya1809:~$ gdb-multiarch -x gdbinit
GNU gdb (Ubuntu 8.1-0ubuntu3) 8.1.0.20180409-git
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word".
Breakpoint 1 at 0x10980: file fileoperations.c, line 24.
Reading /lib/ld-uClibc.so.0 from remote target...
warning: File transfers from remote targets can be slow. Use "set sysroot" to
Reading /lib/ld-uClibc.so.0 from remote target...
Reading /lib//libc.so.0 from remote target...

Breakpoint 1, main () at fileoperations.c:24
24          char *str = NULL;
(gdb) n
26          printf("The following program demonstrates various file operation
(gdb) n
29              printf("\n");
(gdb) n
30              printf("Select operation number to proceed:\n");
(gdb) n
31              printf("1. Print to standard out an interesting string using
```

3. Create a Kernel Module

Reference : http://derekmolloy.ie/writing-a-linux-kernel-module-part-1-introduction/, https://www.oreilly.com/library/view/linux-device-drivers/0596000081/ch06s05.html, Linux Kernel Development, Chapter 11, https://www.tldp.org/LDP/lkmpg/2.6/html/x323.html, https://qnaplus.com/how-to-implement-periodic-timer-in-linux-kernel/,

The following program creates a kernel module using the kernel timer functions. The program contains a timer handler call back function , an init function and an exit function. The init function adds a timer, modifies the timer for starting it and sets the timer up to call the call back function on timeout. The call back function prints the name and the number of times it has been called and remodifies the timer to reset the count again. On doing insmod the module gets added to the kernel and the init function starts to call the call back function(Timer_Handler in this case) periodically depending on the timeout(default or manual) and we can check this when we do dmesg. When we remove the module using rmmod, the exit function is executed this deletes the timer and prints the exit message.

The shreya_timerMod.c file along with the makefile is in Q3 folder of HW3.

To create a kernel module, I have implemented the following steps:

1. Create an out of tree c program (shreya_timerMod.c) following the rules and regulations for making kernel modules as given in the links above.
2. Make the .c file for BBG.(Makefile Reference: MELP book topic- compiling kernel modules)
3. Scp the .ko file generated to the BBG.
4. Scp the timerscript included in the Q3 repository to the BBG(reference: https://stackoverflow.com/questions/13890789/convert-dmesg-timestamp-to-custom-date-format)

```
shreya1809@shreya1809:~/ECEN5013_AESD-S19/Homeworks/HW3/Q3$ ls
built-in.o      Module.symvers      shreya_timerMod.mod.c    shreya_timerMod.o
Makefile        shreya_timerMod.c   shreya_timerMod.mod.dwo
modules.order   shreya_timerMod.ko  shreya_timerMod.mod.o
shreya1809@shreya1809:~/ECEN5013_AESD-S19/Homeworks/HW3/Q3$ sudo scp shreya_time
rMod.ko root@10.0.0.16:/home
root@10.0.0.16's password:
shreya_timerMod.ko                            100%   20KB   1.2MB/s   00:00
shreya1809@shreya1809:~/ECEN5013_AESD-S19/Homeworks/HW3/Q3$ sudo scp timescript.
sh  root@10.0.0.16:/home
root@10.0.0.16's password:
timescript.sh                                 100%  570   150.7KB/s   00:00
```

5. In BBG set date and time manually with #date –set=”20190215814” for 2019, feb, 15, 8:14 pm.
6. For adding a module with default params -> #insmod shreya_timerMod.ko

```
# insmod shreya_timerMod.ko
# lsmod | grep shreya_timerMod
shreya_timerMod        16384  0
#
```

7. For adding a module with different params -> #insmod shreya_timerMod.ko name=SHREYA timeout=5000 ,where timeout is in milliseconds.

```
# cd ../../home
# insmod shreya_timerMod.ko name=SHREYA  timeout=5000
# lsmod
Module                  Size  Used by    Tainted: G
shreya_timerMod        16384  0
cfg80211              610304  0
```

8. To confirm the module has been loaded -> #lsmod | grep “shreya_timerMod”
9. To print the information about the module -> #modinfo shreya_timerMod.ko

```
shreya1809@shreya1809:~/ECEN5013_AESD-S19/Homeworks/HW3/Q3$ modinfo shreya_timer
Mod.ko
filename:       /home/shreya1809/ECEN5013_AESD-S19/Homeworks/HW3/Q3/shreya_timer
Mod.ko
version:        0.1
description:    A simple timer based Linux driver for the BBG.
author:         Shreya Chakraborty
license:        GPL
srcversion:     0284F39A932C749ED77C468
depends:
name:           shreya_timerMod
vermagic:       4.14.40 SMP mod_unload modversions ARMv6 p2v8
parm:           name:The name to display in /var/log/kern.log (charp)
parm:           timeout:The period in miliseconds (ulong)
```

10. To observer the module log-> #bash timerscript.sh

```
# bash timescript.sh
[Fri Feb 15 09:08:22 2019] Timer Handler called
[Fri Feb 15 09:08:22 2019]  Name: shreya
[Fri Feb 15 09:08:22 2019]  No of times : 467
[Fri Feb 15 09:08:23 2019] Timer Handler called
[Fri Feb 15 09:08:23 2019]  Name: shreya
[Fri Feb 15 09:08:23 2019]  No of times : 468
[Fri Feb 15 09:08:23 2019] Timer Handler called
[Fri Feb 15 09:08:23 2019]  Name: shreya
[Fri Feb 15 09:08:23 2019]  No of times : 469
[Fri Feb 15 09:08:24 2019] Timer Handler called
[Fri Feb 15 09:08:24 2019]  Name: shreya
[Fri Feb 15 09:08:24 2019]  No of times : 470
[Fri Feb 15 09:08:24 2019] Timer Handler called
```

11. To remove the module -> #rmmod Shreya_timerMod.ko

```
# rmmod shreya_timerMod.ko
# lsmod
Module                  Size  Used by    Tainted: G
cfg80211              610304  0
```

Proof of Execution:

```
# insmod shreya_timerMod.ko name=SHREYA  timeout=5000
# dmesg -c
[193569.302925] shreya_timerMod inserted into kernel.Function mytimer_init
[193569.310133] The timeout is : 5000 milliseconds
```

```
# bash timescript.sh
[Fri Feb 15 09:37:19 2019] Timer Handler called
[Fri Feb 15 09:37:19 2019]  Name: SHREYA
[Fri Feb 15 09:37:19 2019]  No of times : 5
[Fri Feb 15 09:37:24 2019] Timer Handler called
[Fri Feb 15 09:37:24 2019]  Name: SHREYA
[Fri Feb 15 09:37:24 2019]  No of times : 6
[Fri Feb 15 09:37:29 2019] Timer Handler called
[Fri Feb 15 09:37:29 2019]  Name: SHREYA
[Fri Feb 15 09:37:29 2019]  No of times : 7
[Fri Feb 15 09:37:34 2019] Timer Handler called
[Fri Feb 15 09:37:34 2019]  Name: SHREYA
[Fri Feb 15 09:37:34 2019]  No of times : 8
[Fri Feb 15 09:37:39 2019] Timer Handler called
[Fri Feb 15 09:37:39 2019]  Name: SHREYA
[Fri Feb 15 09:37:39 2019]  No of times : 9
```

```
# rmmod shreya_timerMod.ko
# lsmod
Module                  Size  Used by    Tainted: G
cfg80211              610304  0
# dmesg
[193512.727799] Timer Handler called
[193512.727799]  Name: SHREYA
[193512.727799]  No of times : 35
[193517.767833] Timer Handler called
[193517.767833]  Name: SHREYA
[193517.767833]  No of times : 36
[193522.745538] shreya_timerMod exited from kernel. Function mytimer_exit
#
```

Q4. Data Structures

References:

https://www.geeksforgeeks.org/c-program-bubble-sort-linked-list/

https://www.sanfoundry.com/c-program-find-occurences-all-elements-linked-list/

https://kernelnewbies.org/FAQ/LinkedLists

https://www.programering.com/a/MDN0gDNwATM.html

NOTE: The code for the following is giben in Homeworks/HW3/Q4 of my repository. Animal_type param is case sensitive. Enter animals starting with uppercase char eg Dog,Camel,Cat… etc

Gist of the Problem:

In this problem, generate 2 list, Ecosystem list and Filtered list. Ecosystem list creates a node for all entires, sorts the entire 50 animal seed array to find duplicates and remove them and keep count of the repetition and prints the animal list in alphabetical order along with the count of occurrence. The filtered list takes in module params from the user which can be both animal_type, count greater than, either or none. Based on that it creates another list with the filtered entries of animals.

Data types Used:

Set 1 Ecosystem List: **Link List**

Set 2 Filtered List: **Link List**

Seed Array Entries displayed:

```
[Sun Feb 17 07:52:20 2019] linklist mod entered from kernel. Function mylinklist_init
[Sun Feb 17 07:52:20 2019] The seed array:
[Sun Feb 17 07:52:20 2019] Fox          1
[Sun Feb 17 07:52:20 2019] Bear         1
[Sun Feb 17 07:52:20 2019] Hawk         1
[Sun Feb 17 07:52:20 2019] Camel                1
[Sun Feb 17 07:52:20 2019] Kangaroo             1
[Sun Feb 17 07:52:20 2019] Turtle               1
[Sun Feb 17 07:52:20 2019] Turkey               1
[Sun Feb 17 07:52:20 2019] Hen          1
[Sun Feb 17 07:52:20 2019] Bear         1
[Sun Feb 17 07:52:20 2019] Deer         1
[Sun Feb 17 07:52:20 2019] Shark                1
[Sun Feb 17 07:52:20 2019] Crow         1
[Sun Feb 17 07:52:20 2019] Dog          1
[Sun Feb 17 07:52:20 2019] Beaver               1
[Sun Feb 17 07:52:20 2019] Dog          1
[Sun Feb 17 07:52:20 2019] Snake                1
[Sun Feb 17 07:52:20 2019] Moose                1
[Sun Feb 17 07:52:20 2019] Panther              1
[Sun Feb 17 07:52:20 2019] Beaver               1
[Sun Feb 17 07:52:20 2019] Parrot               1
[Sun Feb 17 07:52:20 2019] Camel                1
[Sun Feb 17 07:52:20 2019] Elephant             1
[Sun Feb 17 07:52:20 2019] Giraffe              1
[Sun Feb 17 07:52:20 2019] Rat          1
[Sun Feb 17 07:52:20 2019] Wolf         1
[Sun Feb 17 07:52:20 2019] Lion         1
[Sun Feb 17 07:52:20 2019] Monkey               1
[Sun Feb 17 07:52:20 2019] Kangaroo             1
[Sun Feb 17 07:52:20 2019] Snake                1
[Sun Feb 17 07:52:20 2019] Aligator             1
[Sun Feb 17 07:52:20 2019] Parrot               1
[Sun Feb 17 07:52:20 2019] Chicken              1
[Sun Feb 17 07:52:20 2019] Turtle               1
```

```
[Sun Feb 17 07:52:20 2019] Turtle                1
[Sun Feb 17 07:52:20 2019] Camel                 1
[Sun Feb 17 07:52:20 2019] Tiger                 1
[Sun Feb 17 07:52:20 2019] Wolf          1
[Sun Feb 17 07:52:20 2019] Wolf          1
[Sun Feb 17 07:52:20 2019] Rabbit                1
[Sun Feb 17 07:52:20 2019] Dog           1
[Sun Feb 17 07:52:20 2019] Turtle                1
[Sun Feb 17 07:52:20 2019] Cow           1
[Sun Feb 17 07:52:20 2019] Dog           1
[Sun Feb 17 07:52:20 2019] Zebra                 1
[Sun Feb 17 07:52:20 2019] Sheep                 1
[Sun Feb 17 07:52:20 2019] Tiger                 1
[Sun Feb 17 07:52:20 2019] Rabbit                1
[Sun Feb 17 07:52:20 2019] Mouse                 1
[Sun Feb 17 07:52:20 2019] Cat           1
[Sun Feb 17 07:52:20 2019] Dog           1
[Sun Feb 17 07:52:20 2019] Lion          1
[Sun Feb 17 07:52:20 2019] Total animal Count :50
```

Set1:

1. **Alphabetically Sorted Ecosystem List**

```
[Sun Feb 17 07:52:20 2019] ECOSYSTEM LIST
[Sun Feb 17 07:52:20 2019] Aligator              1
[Sun Feb 17 07:52:20 2019] Bear          2
[Sun Feb 17 07:52:20 2019] Beaver                2
[Sun Feb 17 07:52:20 2019] Camel                 3
[Sun Feb 17 07:52:20 2019] Cat           1
[Sun Feb 17 07:52:20 2019] Chicken               1
[Sun Feb 17 07:52:20 2019] Cow           1
[Sun Feb 17 07:52:20 2019] Crow          1
[Sun Feb 17 07:52:20 2019] Deer          1
[Sun Feb 17 07:52:20 2019] Dog           5
[Sun Feb 17 07:52:20 2019] Elephant              1
[Sun Feb 17 07:52:20 2019] Fox           1
[Sun Feb 17 07:52:20 2019] Giraffe               1
[Sun Feb 17 07:52:20 2019] Hawk          1
[Sun Feb 17 07:52:20 2019] Hen           1
[Sun Feb 17 07:52:20 2019] Kangaroo              2
[Sun Feb 17 07:52:20 2019] Lion          2
[Sun Feb 17 07:52:20 2019] Monkey                1
[Sun Feb 17 07:52:20 2019] Moose                 1
[Sun Feb 17 07:52:20 2019] Mouse                 1
[Sun Feb 17 07:52:20 2019] Panther               1
[Sun Feb 17 07:52:20 2019] Parrot                2
[Sun Feb 17 07:52:20 2019] Rabbit                2
[Sun Feb 17 07:52:20 2019] Rat           1
[Sun Feb 17 07:52:20 2019] Shark                 1
[Sun Feb 17 07:52:20 2019] Sheep                 1
[Sun Feb 17 07:52:20 2019] Snake                 2
[Sun Feb 17 07:52:20 2019] Tiger                 2
[Sun Feb 17 07:52:20 2019] Turkey                1
[Sun Feb 17 07:52:20 2019] Turtle                3
[Sun Feb 17 07:52:20 2019] Wolf          3
[Sun Feb 17 07:52:20 2019] Zebra                 1
[Sun Feb 17 07:52:20 2019] Total unique animal Count :32
```

2. **Ecosystem no of nodes after Sort:**

```
[Sun Feb 17 07:52:20 2019] Total unique animal Count :32
```

3. **The total amount of memory dynamically allocated for nodes**

```
[Sun Feb 17 07:52:21 2019] Ecosystem List dynamic memory allocated: 512 bytes
```

SET2:

The following screenshots contain the following -> filter criteria, contents of filtered list, number of nodes, memory allocated and deallocated, time to insert & remove.

1. **No params:**

Proof of Execution

```
[Sun Feb 17 08:29:11 2019] FILTERED LIST. Animal_type=(null), Count>=0
[Sun Feb 17 08:29:11 2019] Zebra                1
[Sun Feb 17 08:29:11 2019] Wolf         3
[Sun Feb 17 08:29:11 2019] Turtle               3
[Sun Feb 17 08:29:11 2019] Turkey               1
[Sun Feb 17 08:29:11 2019] Tiger                2
[Sun Feb 17 08:29:11 2019] Snake                2
[Sun Feb 17 08:29:11 2019] Sheep                1
[Sun Feb 17 08:29:11 2019] Shark                1
[Sun Feb 17 08:29:11 2019] Rat          1
[Sun Feb 17 08:29:11 2019] Rabbit               2
[Sun Feb 17 08:29:11 2019] Parrot               2
[Sun Feb 17 08:29:11 2019] Panther              1
[Sun Feb 17 08:29:11 2019] Mouse                1
[Sun Feb 17 08:29:11 2019] Moose                1
[Sun Feb 17 08:29:11 2019] Monkey               1
[Sun Feb 17 08:29:11 2019] Lion         2
[Sun Feb 17 08:29:11 2019] Kangaroo             2
[Sun Feb 17 08:29:11 2019] Hen          1
[Sun Feb 17 08:29:11 2019] Hawk         1
[Sun Feb 17 08:29:11 2019] Giraffe              1
[Sun Feb 17 08:29:11 2019] Fox          1
[Sun Feb 17 08:29:11 2019] Elephant             1
[Sun Feb 17 08:29:11 2019] Dog          5
[Sun Feb 17 08:29:11 2019] Deer         1
[Sun Feb 17 08:29:11 2019] Crow         1
[Sun Feb 17 08:29:11 2019] Cow          1
[Sun Feb 17 08:29:11 2019] Chicken              1
[Sun Feb 17 08:29:11 2019] Cat          1
[Sun Feb 17 08:29:11 2019] Camel                3
[Sun Feb 17 08:29:11 2019] Beaver               2
[Sun Feb 17 08:29:11 2019] Bear         2
[Sun Feb 17 08:29:11 2019] Aligator             1
[Sun Feb 17 08:29:11 2019] Total animal Count :32
[Sun Feb 17 08:29:11 2019] Ecosystem List dynamic memory allocated: 512 bytes
[Sun Feb 17 08:29:11 2019] Filtered List dynamic memory allocated: 512 bytes
[Sun Feb 17 08:29:11 2019] Total dynamic memory allocated: 1024 bytes
[Sun Feb 17 08:29:11 2019] Time to insert the module:309756 us
[Sun Feb 17 08:29:14 2019] Ecosystem List dynamic memory deallocated: 512 bytes
[Sun Feb 17 08:29:14 2019] Filtered List dynamic memory deallocated: 512 bytes
[Sun Feb 17 08:29:14 2019] Total dynamic memory deallocated: 1024 bytes
[Sun Feb 17 08:29:14 2019] Time to exit the module:61 us
[Sun Feb 17 08:29:14 2019] Linklist mod exited from kernel. Function mylinklist_exit
```

2. **Only Animal_type Param**

Proof of execution

```
# sudo insmod data_structMod.ko animal_type=Dog
# sudo rmmod data_structMod.ko
# bash timescript.sh
[Sun Feb 17 08:31:51 2019] linklist mod entered from kernel. Function mylinklist_init
[Sun Feb 17 08:31:51 2019] The seed array:
[Sun Feb 17 08:31:51 2019] Fox          1
[Sun Feb 17 08:31:51 2019] Bear         1
[Sun Feb 17 08:31:51 2019] Hawk         1
[Sun Feb 17 08:31:51 2019] Camel                1
[Sun Feb 17 08:31:51 2019] Total animal Count :50
[Sun Feb 17 08:31:51 2019] ECOSYSTEM LIST
[Sun Feb 17 08:31:51 2019] Aligator             1
[Sun Feb 17 08:31:51 2019] Bear         2
[Sun Feb 17 08:31:51 2019] Beaver               2
[Sun Feb 17 08:31:51 2019] Camel                3
[Sun Feb 17 08:31:51 2019] Cat          1
[Sun Feb 17 08:31:51 2019] Chicken              1
[Sun Feb 17 08:31:51 2019] Cow          1
[Sun Feb 17 08:31:51 2019] Crow         1
[Sun Feb 17 08:31:51 2019] Deer         1
[Sun Feb 17 08:31:51 2019] Dog          5
[Sun Feb 17 08:31:51 2019] Elephant             1
[Sun Feb 17 08:31:51 2019] Fox          1
[Sun Feb 17 08:31:51 2019] Giraffe              1
[Sun Feb 17 08:31:51 2019] Hawk         1
[Sun Feb 17 08:31:51 2019] Hen          1
[Sun Feb 17 08:31:51 2019] Kangaroo             2
[Sun Feb 17 08:31:51 2019] Lion         2
[Sun Feb 17 08:31:51 2019] Monkey               1
[Sun Feb 17 08:31:51 2019] Moose                1
[Sun Feb 17 08:31:51 2019] Mouse                1
[Sun Feb 17 08:31:51 2019] Panther              1
[Sun Feb 17 08:31:51 2019] Parrot               2
[Sun Feb 17 08:31:51 2019] Rabbit               2
[Sun Feb 17 08:31:51 2019] Rat          1
[Sun Feb 17 08:31:51 2019] Shark                1
[Sun Feb 17 08:31:51 2019] Sheep                1
[Sun Feb 17 08:31:51 2019] Snake                2
[Sun Feb 17 08:31:51 2019] Tiger                2
[Sun Feb 17 08:31:51 2019] Turkey               1
[Sun Feb 17 08:31:51 2019] Turtle               3
[Sun Feb 17 08:31:51 2019] Wolf         3
[Sun Feb 17 08:31:51 2019] Zebra                1
[Sun Feb 17 08:31:51 2019] Total unique animal Count :32
[Sun Feb 17 08:31:51 2019] FILTERED LIST. Animal_type=Dog, Count>=0
[Sun Feb 17 08:31:51 2019] Dog          5
[Sun Feb 17 08:31:51 2019] Total animal Count :1
[Sun Feb 17 08:31:51 2019] Ecosystem List dynamic memory allocated: 512 bytes
[Sun Feb 17 08:31:51 2019] Filtered List dynamic memory allocated: 16 bytes
[Sun Feb 17 08:31:51 2019] Total dynamic memory allocated: 528 bytes
[Sun Feb 17 08:31:51 2019] Time to insert the module:237673 us
[Sun Feb 17 08:31:55 2019] Ecosystem List dynamic memory deallocated: 512 bytes
[Sun Feb 17 08:50:30 2019] Filtered List dynamic memory deallocated: 16 bytes
[Sun Feb 17 08:50:30 2019] Total dynamic memory deallocated: 528 bytes
[Sun Feb 17 08:50:30 2019] Time to exit the module:36 us
[Sun Feb 17 08:50:30 2019] Linklist mod exited from kernel. Function mylinklist_exit
#
```

3. **Only count_greater_than param**
   In my case, count greater than param actually shows greater than or equal to.

```
# sudo insmod data_structMod.ko count_greater_than=2
# sudo rmmod data_structMod.ko
# bash timescript.sh
[Sun Feb 17 08:34:48 2019] linklist mod entered from kernel. Function mylinklist_init
[Sun Feb 17 08:34:48 2019] The seed array:
[Sun Feb 17 08:34:48 2019] Fox          1
[Sun Feb 17 08:34:48 2019] Bear         1
[Sun Feb 17 08:34:48 2019] Hawk         1
[Sun Feb 17 08:34:48 2019] Camel                1
[Sun Feb 17 08:34:48 2019] Total unique animal Count :32
[Sun Feb 17 08:34:48 2019] FILTERED LIST. Animal_type=(null), Count>=2
[Sun Feb 17 08:34:48 2019] Wolf         3
[Sun Feb 17 08:34:48 2019] Turtle               3
[Sun Feb 17 08:34:48 2019] Tiger                2
[Sun Feb 17 08:34:48 2019] Snake                2
[Sun Feb 17 08:34:48 2019] Rabbit               2
[Sun Feb 17 08:34:48 2019] Parrot               2
[Sun Feb 17 08:34:48 2019] Lion         2
[Sun Feb 17 08:34:48 2019] Kangaroo              2
[Sun Feb 17 08:34:48 2019] Dog          5
[Sun Feb 17 08:34:48 2019] Camel                3
[Sun Feb 17 08:34:48 2019] Beaver               2
[Sun Feb 17 08:34:48 2019] Bear         2
[Sun Feb 17 08:34:48 2019] Total animal Count :12
[Sun Feb 17 08:34:48 2019] Ecosystem List dynamic memory allocated: 512 bytes
[Sun Feb 17 08:34:48 2019] Filtered List dynamic memory allocated: 192 bytes
[Sun Feb 17 08:34:48 2019] Total dynamic memory allocated: 704 bytes
[Sun Feb 17 08:34:48 2019] Time to insert the module:263516 us
[Sun Feb 17 08:34:51 2019] Ecosystem List dynamic memory deallocated: 512 bytes
[Sun Feb 17 08:34:51 2019] Filtered List dynamic memory deallocated: 192 bytes
[Sun Feb 17 08:34:51 2019] Total dynamic memory deallocated: 704 bytes
[Sun Feb 17 08:34:51 2019] Time to exit the module:44 us
[Sun Feb 17 08:34:51 2019] Linklist mod exited from kernel. Function mylinklist_exit
```

Along with the ecosystem list as in the above case

4. **Both params animal_type and count_greater_than**

```
# sudo insmod data_structMod.ko animal_type=Camel count_greater_than=2
# sudo rmmod data_structMod.ko
# bash timescript.sh
[Sun Feb 17 08:39:27 2019] linklist mod entered from kernel. Function mylinklist_init
[Sun Feb 17 08:39:27 2019] The seed array:
[Sun Feb 17 08:39:27 2019] Fox          1
[Sun Feb 17 08:39:27 2019] Bear         1
[Sun Feb 17 08:39:27 2019] Hawk         1
[Sun Feb 17 08:39:27 2019] Camel                1
[Sun Feb 17 08:39:27 2019] Kangaroo             1
```

Ecosystem List:

```
[Sun Feb 17 08:40:11 2019] Total animal Count :50
[Sun Feb 17 08:40:11 2019] ECOSYSTEM LIST
[Sun Feb 17 08:40:11 2019] Aligator              1
[Sun Feb 17 08:40:11 2019] Bear         2
[Sun Feb 17 08:40:11 2019] Beaver                2
[Sun Feb 17 08:40:11 2019] Camel                 3
[Sun Feb 17 08:40:11 2019] Cat          1
[Sun Feb 17 08:40:11 2019] Chicken               1
[Sun Feb 17 08:40:11 2019] Cow          1
[Sun Feb 17 08:40:11 2019] Crow         1
[Sun Feb 17 08:40:11 2019] Deer         1
[Sun Feb 17 08:40:11 2019] Dog          5
[Sun Feb 17 08:40:11 2019] Elephant              1
[Sun Feb 17 08:40:11 2019] Fox          1
[Sun Feb 17 08:40:11 2019] Giraffe               1
[Sun Feb 17 08:40:11 2019] Hawk         1
[Sun Feb 17 08:40:11 2019] Hen          1
[Sun Feb 17 08:40:11 2019] Kangaroo              2
[Sun Feb 17 08:40:11 2019] Lion         2
[Sun Feb 17 08:40:11 2019] Monkey                1
[Sun Feb 17 08:40:11 2019] Moose                 1
[Sun Feb 17 08:40:11 2019] Mouse                 1
[Sun Feb 17 08:40:11 2019] Panther               1
[Sun Feb 17 08:40:11 2019] Parrot                2
[Sun Feb 17 08:40:11 2019] Rabbit                2
[Sun Feb 17 08:40:11 2019] Rat          1
[Sun Feb 17 08:40:11 2019] Shark                 1
[Sun Feb 17 08:40:11 2019] Sheep                 1
[Sun Feb 17 08:40:11 2019] Snake                 2
[Sun Feb 17 08:40:11 2019] Tiger                 2
[Sun Feb 17 08:40:11 2019] Turkey                1
[Sun Feb 17 08:40:11 2019] Turtle                3
[Sun Feb 17 08:40:11 2019] Wolf         3
[Sun Feb 17 08:40:11 2019] Zebra                 1
[Sun Feb 17 08:40:11 2019] Total unique animal Count :32
```

Filtered List:

```
[Sun Feb 17 08:40:11 2019] FILTERED LIST. Animal_type=Camel, Count>=2
[Sun Feb 17 08:40:11 2019] Camel                3
[Sun Feb 17 08:40:11 2019] Total animal Count :1
[Sun Feb 17 08:40:11 2019] Ecosystem List dynamic memory allocated: 512 bytes
[Sun Feb 17 08:40:11 2019] Filtered List dynamic memory allocated: 16 bytes
[Sun Feb 17 08:40:11 2019] Total dynamic memory allocated: 528 bytes
[Sun Feb 17 08:40:11 2019] Time to insert the module:237740 us
[Sun Feb 17 08:40:15 2019] Ecosystem List dynamic memory deallocated: 512 bytes
[Sun Feb 17 08:40:15 2019] Filtered List dynamic memory deallocated: 16 bytes
[Sun Feb 17 08:40:15 2019] Total dynamic memory deallocated: 528 bytes
[Sun Feb 17 08:40:15 2019] Time to exit the module:35 us
[Sun Feb 17 08:40:15 2019] Linklist mod exited from kernel. Function mylinklist_exit
# 
```