

## Assignment 1

### RB Tree and Dynamic Probe in Linux Kernel

## Introduction

Implement a Kernel Module of RB Tree and have Dynamic Probe (kprobe) on the operations of RB Tree.

Part 1 of the assignment is based on inserting and deleting values on RBTree module `rbt530_dev`. The balanced BST is based on key value and data is user specific.

Insert:- Whenever a write happens, a new node is created and inserted on the RB Tree, provided it has a unique key. If the key already exists on the tree, the node is replaced with a new node. If data is 0, the node with key value provided is being deleted.

Read:-The RB Tree can be read in ascending or descending order based on the user's choice via `ioctl` call. If the argument of the `ioctl` is "ascending", the RB Tree content is displayed in ascending order, if "descending", its displayed in descending order. For any other values the `ioctl` implementation will defaults to "descending"

Part 2 of the assignment is the ability to put dynamic probe on the RB Tree module. A module "RBprobe" is created to accomplish this task. Users can dynamically give the offset or symbol name to probe the RB Tree from the user interface along with 1 or 0 to register or deregister the address to be probed. The read function of the module reads the RB Tree content and displays it into the user console.

To test the functions of RB Tree and Probe, two instances of RB Tree module is created , with two different minor numbers for the same major number of `"rbt530_dev"`, namely `"rbt530_dev1"` and `"rbt530_dev2"`. Each instance of rb tree is operated by two threads. Apart from these four threads, fifth thread is created for the functionality of probe i.e take the address and flag to register/deregister and display the content of the tree in user space.

# Mandatory Requirement:

- RBT\_530 module can be independently compiled and run with the Makefile in Part1 folder along with rbt\_tester\_probe application.
- RBProbe module requires RBT\_530 module and rbt\_tester\_probe application to probe the RBTree instructions. However the Makefile in Part2 is meant to compile RTprobe.c independently.

## Content of the Assignments

### 1. C files

/\*User space test program for two modules. Executable is named rbt\_tester\_probe \*/  
Main.c  
/\*Driver module program for Dynamic Probe \*/  
RBprobe.c  
/\*Driver module program for RBTree 530 \*/  
RBT\_530.c

### 2. Header files

/\*ioctl declaration for setting up ascending and descending order for RBTree\*/  
RBT\_530.h  
/\*shares the definition of Tree Nodes structure between tree and probe modules\*/  
RBT\_Nodes.h

### 3. Shell scripts

/\*makes modules and tester app and copies the files to /tftpboot/ if they are compiled from same directory.

Buildscript.sh

/\*download \*.ko from /tftpboot/ and install the modules using insmod \*/

Install.sh

/\*uninstall the modules from the kernel using rmmod command \*/

Uninstall.sh

/\*uninstalls, installs and starts the main program \*/

4. /\*Makefile compiles the C files builds the modules for Galileo gen 2 board \*/  
Makefile. The Makefile in Part1 can include RBprobe with small edit to build from same location.

# Steps

## Compilation and Installation Steps

`/* On Host machine*/ <buildscript.sh>`

1. `make clean`
2. `make` (to compile both kernel module and tester app)
3. `cp *.ko /tftpboot/`
4. `cp rbt_tester_probe /tftpboot`

`/*Target Machine*/ <./uninstall.sh> followed by <install.sh>`

`<edit install.sh to added ascending or descending while executing ./rbt_tester_probe>`

1. `cp /tftpboot/*.ko to desired location`
2. `Lsmod |grep RBT_530`  
if (installed)  
    `Rmmod RBT_530`  
    `Insmod RBProbe`
3. `Lsmod |grep RBprobe`  
if (installed)  
    `Rmmod RBprobe`  
    `Insmod RBProbe`
4. `./rbt_tester_probe <ascending|descending>`

## Helper scripts

`/*Host Machine`

1. Execute `./buildscript.sh` in host machine

`/* Target (Galileo gen 2)`

2. `./install.sh` , downloads the files from `/tftpboot/` and installs the modules and executes tester program
3. `./uninstall.sh` , to uninstall the kernel modules

## Input

1. **RBTree** The key and data are randomly generated in user space and are written to buffer which is then received in kernel module `RTB530`.  
Users have the choice to specify address to probe along with option “ascending” or “descending” to display the tree information in ascending or descending order.

2. KBprobe: KbProbe thread is created and waits for user input to register/deregister probe on RBTree specified by address and a flag. Flag specifies to register or deregister the probe for particular address. Currently the first two probes are hardcoded to insert and read the function of RBTree.  
Users can type either add <address> <flag> to register/deregister or stop to stop the program.

## Processing

1. RBTree: The threads randomly performs insert/read tree and display(dump entire tree) based on flag for ascending or descending.
2. KBProbe: User thread, wait for user to enter address and flag and simultaneously kprobe module performs probe on insert and read from addresses which are currently hardcoded.

## Output

1. RBTree -
  - Operations read/insert/display of RBTree is displayed on the console.
2. From Kprobe
  - Display the information of timestamp when the probe was hit, PID of the thread which executes the command and key and data for which the probe was hit.
3. Dmesg gives the output of the RBTrees module. It shows the content of the RBTrees when the program closes

```
Activities Terminal Fri 14:48
mou@mou-lin: ~/Documents/embedded/final_assign1/backup7/backup
File Edit View Search Terminal Tabs Help
mou@mou-lin: ~/Documents/embedded/final_assign1/backup7/backup x mou@mou-lin: ~/Documents/embedded/final_assign1/backup7/backup x mou@mou-lin: ~/Documents/embedded/final_assign1/backup7/backup x
Device(s) open successful
Device IOCTL call: Mode:ascending
THREAD CREATED FOR RBT
Thread created with ID: 357
Thread created with ID: 358
Thread created with ID: 356
Thread created with ID: 359
[Thread 360 ] Kprobe Thread
Thread 357: Write RBT: [Key=8 Value=193] [Key=8 Value=186] [Key=1 Value=49]
Thread 357: Write RBT: [Key=3 Value=27]
=====
add: To add an address for probing
stop: To exit
=====
Enter add or stop:
=====Probe Result =====
[Timestamp= 429523603923 ] [Address=0xd29592c7 ] [ProcessId=357 ] [Key= 8 Value= 193]
[Timestamp= 429523627850 ] [Address=0xd29592c7 ] [ProcessId=357 ] [Key= 1 Value= 49]
[Timestamp= 429523702742 ] [Address=0xd29592c7 ] [ProcessId=357 ] [Key= 3 Value= 27]
=====
Thread 358: Write RBT: [Key=3 Value=59] [Key=12 Value=126] [Key=1 Value=26]
Thread 358: Random Delete RBT: [Key=5 Value=0]
Thread 356: Write RBT: [Key=12 Value=168] [Key=4 Value=29] [Key=3 Value=130]
[Thread 356 ]: Read Operation on RBT
=====Tree(1)=====
READ FULL TREE
[Key=1 Value=49] [Key=3 Value=130] [Key=4 Value=29] [Key=8 Value=186] [Key=12 Value=168]
=====
[Timestamp= 429529204177 ] [Address=0xd29592c7 ] [ProcessId=358 ] [Key= 3 Value= 59]
[Timestamp= 429529218788 ] [Address=0xd29592c7 ] [ProcessId=358 ] [Key= 12 Value= 126]
[Timestamp= 429529230858 ] [Address=0xd29592c7 ] [ProcessId=358 ] [Key= 1 Value= 26]
[Timestamp= 429529476261 ] [Address=0xd29592c7 ] [ProcessId=356 ] [Key= 12 Value= 168]
[Timestamp= 429529492328 ] [Address=0xd29592c7 ] [ProcessId=356 ] [Key= 4 Value= 29]
[Timestamp= 429640534755 ] [Address=0xd2959110 ] [ProcessId=356 ] [Key= 3 Value= 130]
Thread 359: Write RBT: [Key=8 Value=67] [Key=8 Value=129] [Key=7 Value=22]
[Thread 359 ]: Read Operation on RBT
...
[]
```

Output with RBprobe

```
Activities Terminal Fri 17:51
mou@mou-lin: ~/Documents/embedded/final_assign1/EOSI_Laskar_Moumita_assgn01/Part1

mou@mou-lin: ~/Documents/embedded/final_assign1/EOSI_Laskar_Moumita_assgn01/Part1
>12 [Data]=>22 [Key]=>15 [Data]=>59 [Key]=>17 [Data]=>116 [Key]=>20 [Data]=>24 [Key]=>23 [Data]=>111 [Key]=>26 [Data]=>44 [Key]=>27 [Data]=>122 [Key]=>29 [Data]=>
>99 [Key]=>31 [Data]=>84 [Key]=>32 [Data]=>81 [Key]=>34 [Data]=>48 [Key]=>37 [Data]=>60 [Key]=>38 [Data]=>5 [Key]=>39 [Data]=>6 [Key]=>40 [Data]=>169 [Key]=>
42 [Data]=>169 [Key]=>44 [Data]=>146 [Key]=>45 [Data]=>99 [Key]=>49 [Data]=>103 [Key]=>53 [Data]=>196 [Key]=>55 [Data]=>126 [Key]=>56 [Data]=>19 [Key]=>58 [Data]=>
>165 [Key]=>59 [Data]=>70 [Key]=>61 [Data]=>104 [Key]=>64 [Data]=>97 [Key]=>65 [Data]=>197 [Key]=>66 [Data]=>188 [Key]=>68 [Data]=>77 [Key]=>69 [Data]=>140 [Key]=>
[Key]=>71 [Data]=>62 [Key]=>74 [Data]=>176 [Key]=>82 [Data]=>109 [Key]=>86 [Data]=>136 [Key]=>90 [Data]=>83 [Key]=>91 [Data]=>103 [Key]=>93 [Data]=>86 [Key]=>96
[Data]=>126
[ 8566.061243] -----
[ 8566.102802] rbt530_dev0 is closing
[ 8566.112180] -----
[ 8566.112180] -----For Tree rbt530_dev1-----
[ 8566.120223] [Key]=>1 [Data]=>38 [Key]=>3 [Data]=>22 [Key]=>6 [Data]=>46 [Key]=>8 [Data]=>159 [Key]=>9 [Data]=>76 [Key]=>13 [Data]=>26 [Key]=>14 [Data]=>68 [Key]=>16 [Data]=>121 [Key]=>21 [Data]=>84 [Key]=>22 [Data]=>162 [Key]=>24 [Data]=>67 [Key]=>26 [Data]=>84 [Key]=>27 [Data]=>67 [Key]=>29 [Data]=>125 [Key]=>31 [Data]=>
>99 [Key]=>36 [Data]=>129 [Key]=>37 [Data]=>163 [Key]=>38 [Data]=>140 [Key]=>43 [Data]=>29 [Key]=>44 [Data]=>74 [Key]=>46 [Data]=>14 [Key]=>47 [Data]=>35 [Key]=>49 [Data]=>123 [Key]=>51 [Data]=>68 [Key]=>52 [Data]=>154 [Key]=>54 [Data]=>81 [Key]=>58 [Data]=>94 [Key]=>60 [Data]=>106 [Key]=>61 [Data]=>36 [Key]=>64 [Data]=>10 [Key]=>65 [Data]=>85 [Key]=>66 [Data]=>89 [Key]=>68 [Data]=>34 [Key]=>69 [Data]=>193 [Key]=>71 [Data]=>184 [Key]=>74 [Data]=>4 [Key]=>75 [Data]=>91 [Key]=>76 [Data]=>50 [Key]=>79 [Data]=>67 [Key]=>80 [Data]=>177 [Key]=>82 [Data]=>3 [Key]=>83 [Data]=>22 [Key]=>85 [Data]=>3 [Key]=>88 [Data]=>15 [Key]=>90 [Data]=>196 [Key]=>92 [Data]=>153 [Key]=>94 [Data]=>86 [Key]=>96 [Data]=>182 [Key]=>98 [Data]=>102 [Key]=>100 [Data]=>196
[ 8566.194209] -----
[ 8566.247384] rbt530_dev1 is closing
[ 8939.411452] rbt530_dev0 is opening
[ 8939.41631] rbt530_dev1 is opening
[ 8939.427178] ascending
[ 8939.431051] inside ioctl 1
[ 8953.788662] -----
[ 8953.788662] -----For Tree rbt530_dev0-----
[ 8953.797906] [Key]=>1 [Data]=>2 [Key]=>6 [Data]=>21 [Key]=>7 [Data]=>15 [Key]=>8 [Data]=>159 [Key]=>9 [Data]=>195 [Key]=>10 [Data]=>187 [Key]=>13 [Data]=>90 [Key]=>14 [Data]=>161 [Key]=>16 [Data]=>121 [Key]=>18 [Data]=>30 [Key]=>19 [Data]=>28 [Key]=>21 [Data]=>84 [Key]=>22 [Data]=>42 [Key]=>27 [Data]=>67 [Key]=>33 [Data]=>
>134 [Key]=>36 [Data]=>193 [Key]=>37 [Data]=>60 [Key]=>38 [Data]=>140 [Key]=>40 [Data]=>169 [Key]=>41 [Data]=>149 [Key]=>42 [Data]=>150 [Key]=>43 [Data]=>179 [Key]=>44 [Data]=>11 [Key]=>48 [Data]=>31 [Key]=>49 [Data]=>90 [Key]=>53 [Data]=>196 [Key]=>55 [Data]=>167 [Key]=>56 [Data]=>60 [Key]=>58 [Data]=>81 [Key]=>59 [Data]=>70 [Key]=>60 [Data]=>106 [Key]=>61 [Data]=>36 [Key]=>64 [Data]=>49 [Key]=>65 [Data]=>28 [Key]=>67 [Data]=>103 [Key]=>68 [Data]=>104 [Key]=>69 [Data]=>193 [Key]=>70 [Data]=>26 [Key]=>74 [Data]=>176 [Key]=>75 [Data]=>91 [Key]=>76 [Data]=>50 [Key]=>84 [Data]=>176 [Key]=>85 [Data]=>193 [Key]=>86 [Data]=>22 [Key]=>88 [Data]=>15 [Key]=>90 [Data]=>196 [Key]=>92 [Data]=>153 [Key]=>94 [Data]=>86 [Key]=>96 [Data]=>182 [Key]=>98 [Data]=>102 [Key]=>100 [Data]=>196
>88 [Data]=>15 [Key]=>89 [Data]=>74 [Key]=>90 [Data]=>74 [Key]=>93 [Data]=>86 [Key]=>94 [Data]=>100 [Key]=>95 [Data]=>60 [Key]=>97 [Data]=>167 [Key]=>98 [Data]=>
49 [Key]=>100 [Data]=>196
[ 8953.913552] -----
[ 8953.981996] rbt530_dev0 is closing
[ 8954.000197] -----
[ 8954.000197] -----For Tree rbt530_dev1-----
[ 8954.008097] [Key]=>1 [Data]=>59 [Key]=>2 [Data]=>61 [Key]=>3 [Data]=>22 [Key]=>4 [Data]=>53 [Key]=>6 [Data]=>59 [Key]=>7 [Data]=>195 [Key]=>10 [Data]=>127 [Key]=>14 [Data]=>172 [Key]=>15 [Data]=>71 [Key]=>17 [Data]=>116 [Key]=>18 [Data]=>92 [Key]=>21 [Data]=>167 [Key]=>24 [Data]=>67 [Key]=>27 [Data]=>66 [Key]=>29 [Data]=>
>127 [Key]=>31 [Data]=>62 [Key]=>32 [Data]=>117 [Key]=>33 [Data]=>60 [Key]=>34 [Data]=>85 [Key]=>35 [Data]=>178 [Key]=>36 [Data]=>129 [Key]=>37 [Data]=>163 [Key]=>38 [Data]=>91 [Key]=>43 [Data]=>29 [Key]=>44 [Data]=>173 [Key]=>45 [Data]=>99 [Key]=>49 [Data]=>123 [Key]=>51 [Data]=>68 [Key]=>54 [Data]=>81 [Key]=>55 [Data]=>199 [Key]=>56 [Data]=>19 [Key]=>57 [Data]=>179 [Key]=>58 [Data]=>94 [Key]=>59 [Data]=>73 [Key]=>63 [Data]=>170 [Key]=>65 [Data]=>85 [Key]=>66 [Data]=>147 [Key]=>69 [Data]=>167 [Key]=>71 [Data]=>184 [Key]=>75 [Data]=>126 [Key]=>80 [Data]=>177 [Key]=>82 [Data]=>3 [Key]=>86 [Data]=>136 [Key]=>87 [Data]=>53 [Key]=>88 [Data]=>8 [Key]=>89 [Data]=>184 [Key]=>90 [Data]=>83 [Key]=>96 [Data]=>170 [Key]=>97 [Data]=>179
[ 8954.081823] -----
[ 8954.132961] rbt530_dev1 is closing
...
root@quark:~/rbtree#
```

Output of RBT\_530 module(dmesg output)

## Deliverables:

Folder EOSI\_LASTNAME\_FIRSTNAME\_assgn01

Readme

Part1

- Main.c
- Makefile
- RBT\_530.c
- RBT\_530.h
- RBT\_Nodes.h

- This Makefile creates module RBT\_530.ko for read/write and ioctl operations

## Part2

- RBprobe.c
- Buildscript.sh
- Install.sh
- Uninstall.sh
- Makefile
- This Makefile will is required to have RBT\_530.c in the same folder as there is symbol dependencies.