1. Three Child Processes are created.
2.

```
  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
 1523 paulco    20   0 4820964 362856 141024 S   2.0   3.8   1:15.02 gnome-shell
 2307 paulco    20   0 3384324 377880 156780 S   1.3   4.0   0:50.14 firefox
  209 root      20   0       0      0      0 S   0.7   0.0   0:00.44 jbd2/sda3-8
   97 root       0 -20       0      0      0 I   0.3   0.0   0:01.09 kworker/0:1H-kblockd
  662 root      20   0  269676  18680  15828 S   0.3   0.2   0:01.00 NetworkManager
 1640 paulco    20   0  632440  14156  11676 S   0.3   0.1   0:00.38 xdg-desktop-por
```

3.

```
paulco@paulco-VirtualBox:~$ free -m
               total        used        free      shared  buff/cache   available
Mem:            9279        1166        6713          66        1399        7806
Swap:           2047           0        2047
```

4. Gnome process consumes the most memory. Look at the screenshot for question #2.
5. Firefox has the most memory.
6. Commands:
    a. Apt-get is a command-line tool for handling packages.
    b. Yum is used to install, remove, and update packages from the terminal.
    c. Wget is a free utility for non-interactive download of files from the web.
    d. Gzip reduces the size of the named files using Lempel-ziv coding.
    e. Tar is an archiving utility.
    f. Rar command allows users to create and manage RAR archives, including options for adding, extracting, listing, testing, and repairing archive files.
7.

```
  GNU nano 6.2
#include <stdio.h>
#include <unistd.h>

int main(){

        pid_t pid = fork();
        if(pid == 0){
                //child process
                for(int i = 0; i < 200; i++){
                        printf("I am the  child process\n ");
                }
        }else{
                //parent process
                for(int i = 0; i < 200; i++){
                        printf("I am the parent process\n");
                }
        }
        return 0;
}
```

```
paulco@paulco-VirtualBox:~/tmp/OS/CWU/LabAssignments/lab3$ gcc -g genChild.c -o genChild
paulco@paulco-VirtualBox:~/tmp/OS/CWU/LabAssignments/lab3$ ./genChild
I am the parent process
I am the parent process
I am the parent process
I am the parent process
I am the parent process
I am the parent process
I am the parent process
I am the parent process
I am the parent process
I am the parent process
I am the parent process
I am the parent process
I am the parent process
I am the parent process
I am the parent process
I am the parent process
I am the parent process
I am the parent process
I am the parent process
I am the parent process
I am the parent process
I am the parent process
I am the parent process
I am the parent process
I am the parent process
I am the parent process
I am the parent process
I am the parent process
I am the parent process
I am the parent process
I am the parent process
I am the parent process
I am the parent process
I am the parent process
I am the parent process
I am the parent process
I am the  child process
 I am the  child process
 I am the  child process
 I am the  child process
 I am the  child process
 I am the  child process
 I am the  child process
 I am the  child process
 I am the  child process
 I am the  child process
 I am the  child process
 I am the  child process
 I am the  child process
 I am the  child process
 I am the  child process
 I am the  child process
```

8.

GNU nano 6.2 childWFork.c

```c
#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>
#include <sys/wait.h>
#include <stdlib.h>

int main(){
        pid_t pid = fork();
        char *buf;

        if(pid == 0){
                //Child process
                printf("I am the child procesn\n");
        }else{
                //parent process
                wait(NULL);
        /* When the child is ended , then the parent will continue to execute its code  */
        printf("Child complete \n");
        printf("I am the parent process \n");
        buf = (char *)malloc(100*sizeof(char));
        getcwd(buf,100);
        printf("\n %s \n",buf );
        }

        return 0;
}
```

```
paulco@paulco-VirtualBox:~/tmp/OS/CWU/LabAssignments/lab3$ ./childWFork
I am the child procesn
Child complete
I am the parent process

 /home/paulco/tmp/OS/CWU/LabAssignments/lab3
```

9.

GNU nano 6.2 lab3_9.c

```c
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>

int main (){
        pid_t pid = fork();

        if(pid ==  0){
        //child process
                printf("I am the child  process, my PID is %d, my parent's PID is %d\n", getpid(), getppid());
        }else{
        //parent  process
                printf("I am the Parent, my PID is %d, my child's PID is %d\n ", getpid(), pid);
        }
        return 0;
}
```

```
paulco@paulco-VirtualBox:~/tmp/OS/CWU/LabAssignments/lab3$ ./lab3_9
I am the parent process, my PID is 2331
 My childs process is 2332
 I am the child process, my PID is 2332
 my parent process is 1350  paulco@paulco-VirtualBox:~/tmp/OS/CWU/LabAssignments/lab3$ nano lab3_9.c
paulco@paulco-VirtualBox:~/tmp/OS/CWU/LabAssignments/lab3$
```