Faculty of Information Engineering and Technology German University in Cairo



CSEN 602: Operating Systems – Spring '18 Mini-project I – Report

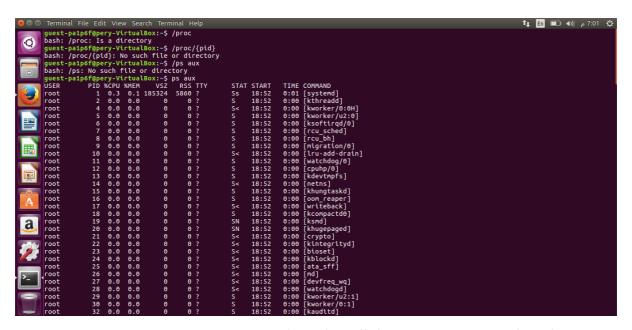
Team ID:22 {22}

Perihan Mohamed Kamel 37-4034 Seif Eldin Ahmed AbdElMonsef 37-5628

1. Part I - System Calls

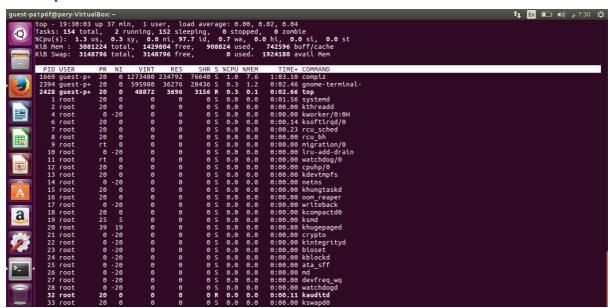
1.1 System Call #1:

- System call command: *ps aux*
- Output screenshot:



• Brief description of the output: it showed us all the users' processes, show the process listed in a user oriented fashion and show us the processes not just attached to terminals but such as services as well, User for user owning the process, PID for process id ,%cpu it is the cpu time used divided by the time the process has ben running,%MEM is the ratio of the process's resident size to the physical memory of the machine, VSZ is the virtual memory usage of entire process, RSS is the resident set size, TTY is the Terminal, STAT is multi character process state, START is starting time or date of process, TIME is cumulative CPU time and COMMAND is the command with all it arguments.

- 1.2 System Call #2:
- System call command: *top*
- Output screenshot:



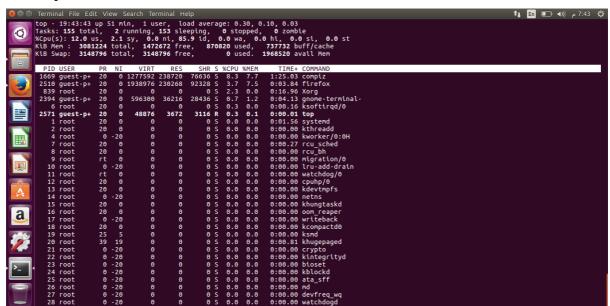
Brief description of the output: The top program provides a dynamic real-time view of a running system. It can display system summary information as well as a list of processes or threads currently being managed by the Linux kernel. The types of system summary information shown and the types, order and size of information displayed for processes are all user configurable and that configuration can be made persistent across restarts.

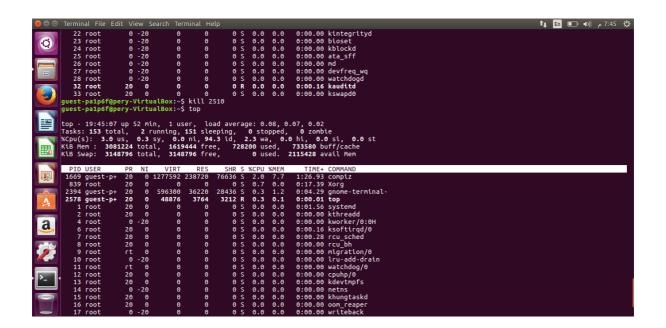
1.3 System Call #3:

• System call command: *top*

Kill 2510 top

• Output screenshot:





• Brief description of the output: the firefox process was found in the process table from the TOP system call and when I call Kill and the ID of the firefox and then called TOP again the process of firefox is not found anymore because of the kill system call.

2. Part II - Directory Duplication Process in C

```
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
#include <stdio.h>
#include <stdib.h>

/*
Other include statements may be needed!
*/
int main() {

char path[50];
printf("Please enter your path\n");
scanf("%s",path);
char name[50];
printf("Please enter the directory name\n");
```

```
scanf("%s",name);
char pa[50];
strcat(pa,"/");
// char * directoryName = argv[1]; /* Directory name to be created */
struct stat st = \{0\};
if (stat(pa, &st) == -1) {
mkdir(pa, 0700);
else
{ int a;
  char final[300];
  for(a = 1; a < 20; a++){
  char s[200];
  char *str = pa;
char buf[30];
sprintf(buf,"%d",a);
printf("%s",final);
if (stat(final, &st) == -1) {
```

CSEN 602 - OPERATING SYSTEMS- MINI-PROJECT I REPORT

```
mkdir(final, 0700);
break;
}
}

/*
Rest of implementation goes here
*/
```

Make sure to convert this file to PDF