

Operating Systems Assignment - 1

Team:

Srinivas Piskala Ganesh Babu (spg349 – N13138339)

Nanda Kishore Kalidindi (nkk263 -)

Figure out how to list running processes on your computer. (ps -ef will do it on UNIX-based computers.) How many processes are running? Choose ten of them, for which you have no clue what they are doing. Figure out what it is they do!

- Linux Operating System:

| UID | PID | PPID | C | STIME | TTY | TIME | CMD |
|------|-----|------|---|-------|-----|----------|----------------|
| root | 1 | 0 | 0 | 11:05 | ? | 00:00:01 | /sbin/init |
| root | 2 | 0 | 0 | 11:05 | ? | 00:00:00 | [kthreadd] |
| root | 3 | 2 | 0 | 11:05 | ? | 00:00:00 | [ksoftirqd/0] |
| root | 5 | 2 | 0 | 11:05 | ? | 00:00:00 | [kworker/0:0H] |
| root | 7 | 2 | 0 | 11:05 | ? | 00:00:06 | [rcu sched] |
| root | 8 | 2 | 0 | 11:05 | ? | 00:00:00 | [rcu bh] |
| root | 9 | 2 | 0 | 11:05 | ? | 00:00:00 | [migration/0] |
| root | 10 | 2 | 0 | 11:05 | ? | 00:00:00 | [watchdog/0] |
| root | 11 | 2 | 0 | 11:05 | ? | 00:00:00 | [watchdog/1] |
| root | 12 | 2 | 0 | 11:05 | ? | 00:00:00 | [migration/1] |
| root | 13 | 2 | 0 | 11:05 | ? | 00:00:00 | [ksoftirqd/1] |
| root | 15 | 2 | 0 | 11:05 | ? | 00:00:00 | [kworker/1:0H] |
| root | 16 | 2 | 0 | 11:05 | ? | 00:00:00 | [watchdog/2] |
| root | 17 | 2 | 0 | 11:05 | ? | 00:00:00 | [migration/2] |
| root | 18 | 2 | 0 | 11:05 | ? | 00:00:00 | [ksoftirqd/2] |
| root | 20 | 2 | 0 | 11:05 | ? | 00:00:00 | [kworker/2:0H] |
| root | 21 | 2 | 0 | 11:05 | ? | 00:00:00 | [watchdog/3] |
| root | 22 | 2 | 0 | 11:05 | ? | 00:00:00 | [migration/3] |
| root | 23 | 2 | 0 | 11:05 | ? | 00:00:00 | [ksoftirqd/3] |
| root | 25 | 2 | 0 | 11:05 | ? | 00:00:00 | [kworker/3:0H] |
| root | 26 | 2 | 0 | 11:05 | ? | 00:00:00 | [kdevtmpfs] |
| root | 27 | 2 | 0 | 11:05 | ? | 00:00:00 | [netns] |
| root | 28 | 2 | 0 | 11:05 | ? | 00:00:00 | [perf] |
| root | 29 | 2 | 0 | 11:05 | ? | 00:00:00 | [khungtaskd] |
| root | 30 | 2 | 0 | 11:05 | ? | 00:00:00 | [writeback] |
| root | 31 | 2 | 0 | 11:05 | ? | 00:00:01 | [ksmd] |
| root | 32 | 2 | 0 | 11:05 | ? | 00:00:04 | [khugepaged] |
| root | 33 | 2 | 0 | 11:05 | ? | 00:00:00 | [crypto] |
| root | 34 | 2 | 0 | 11:05 | ? | 00:00:00 | [kintegrityd] |
| root | 35 | 2 | 0 | 11:05 | ? | 00:00:00 | [bioset] |
| root | 36 | 2 | 0 | 11:05 | ? | 00:00:00 | [kblockd] |
| root | 37 | 2 | 0 | 11:05 | ? | 00:00:00 | [ata_sff] |
| root | 38 | 2 | 0 | 11:05 | ? | 00:00:00 | [md] |
| root | 39 | 2 | 0 | 11:05 | ? | 00:00:00 | [devfreq wq] |
| root | 43 | 2 | 0 | 11:05 | ? | 00:00:00 | [kswapd0] |

a. KThread: Kernel Thread

Management of Hardware. High Priority set as they are handled by the kernel.

b. WatchDog

The Daemon that monitors and tells the kernel that the system is working fine. If it stops then the kernel resets.

c. KWorker – Kernel Worker Thread

Perform the processing of the Kernel especially in cases of Interrupts, Timers. IO

- d. Migration** — Distribution of work load across the CPU Cores
They move kernel threads across cpu cores for processing.

- **MAC OS:**

| UID | PID | PPID | C | STIME | TTY | TIME | CMD |
|---|-----|------|---|---------|-----|---------|---|
| 0 | 1 | 0 | 0 | 12:59AM | ?? | 0:02.82 | /sbin/launchd |
| 0 | 34 | 1 | 0 | 12:59AM | ?? | 0:00.69 | /usr/libexec/UserEventAgent (System) |
| 0 | 35 | 1 | 0 | 12:59AM | ?? | 0:00.21 | /usr/sbin/syslogd |
| 0 | 37 | 1 | 0 | 12:59AM | ?? | 0:00.03 | /System/Library/PrivateFrameworks/Uninstall.framework/Resources/uninstalld |
| 0 | 38 | 1 | 0 | 12:59AM | ?? | 0:01.40 | /usr/libexec/kextd |
| 0 | 39 | 1 | 0 | 12:59AM | ?? | 0:07.37 | |
| /System/Library/Frameworks/CoreServices.framework/Versions/A/Frameworks/FSEvents.framework/Versions/A/Support/fseventsd | | | | | | | |
| 0 | 41 | 1 | 0 | 12:59AM | ?? | 0:00.08 | /System/Library/PrivateFrameworks/MediaRemote.framework/Support/mediaremoted |
| 55 | 44 | 1 | 0 | 12:59AM | ?? | 0:00.11 | /System/Library/CoreServices/appleeventsd --server |
| 0 | 45 | 1 | 0 | 12:59AM | ?? | 0:00.57 | /usr/libexec/configd |
| 0 | 46 | 1 | 0 | 12:59AM | ?? | 0:00.16 | /System/Library/CoreServices/powerd.bundle/powerd |
| 0 | 47 | 1 | 0 | 12:59AM | ?? | 0:01.05 | /usr/libexec/mobileassetd |
| 0 | 48 | 1 | 0 | 12:59AM | ?? | 0:00.02 | |
| /System/Library/PrivateFrameworks/InstallerDiagnostics.framework/Versions/A/Resources/installerdiagwatcher | | | | | | | |
| 0 | 51 | 1 | 0 | 12:59AM | ?? | 0:00.48 | /usr/libexec/logd |
| 0 | 55 | 1 | 0 | 12:59AM | ?? | 0:01.21 | /usr/libexec/airportd |
| 0 | 57 | 1 | 0 | 12:59AM | ?? | 0:00.08 | /usr/libexec/warmd |
| 0 | 58 | 1 | 0 | 12:59AM | ?? | 0:03.61 | |
| /System/Library/Frameworks/CoreServices.framework/Frameworks/Metadata.framework/Support/mds | | | | | | | |
| 240 | 63 | 1 | 0 | 12:59AM | ?? | 0:00.05 | /System/Library/CoreServices/iconservicesd |
| 0 | 64 | 1 | 0 | 12:59AM | ?? | 0:00.05 | /System/Library/CoreServices/iconservicesagent |
| 0 | 66 | 1 | 0 | 12:59AM | ?? | 0:00.19 | /usr/libexec/diskarbitrationd |
| 0 | 68 | 1 | 0 | 12:59AM | ?? | 0:00.71 | /usr/libexec/coreduetd |
| 0 | 69 | 1 | 0 | 12:59AM | ?? | 0:00.04 | /usr/libexec/wdhelper |
| 0 | 72 | 1 | 0 | 12:59AM | ?? | 0:00.12 | /System/Library/CoreServices/ionodecache -k /var/db/ionodecache.json |
| 0 | 74 | 1 | 0 | 12:59AM | ?? | 0:02.51 | /usr/libexec/opaidd |
| 0 | 75 | 1 | 0 | 12:59AM | ?? | 0:00.06 | /usr/sbin/wirelessproxd |
| 0 | 77 | 1 | 0 | 12:59AM | ?? | 0:00.80 | /System/Library/PrivateFrameworks/ApplePushService.framework/apspd |
| 0 | 78 | 1 | 0 | 12:59AM | ?? | 0:00.03 | /System/Library/PrivateFrameworks/Noticeboard.framework/Versions/A/Resources/nbstd |
| 0 | 79 | 1 | 0 | 12:59AM | ?? | 0:00.92 | /System/Library/CoreServices/launchservicesd |
| 213 | 80 | 1 | 0 | 12:59AM | ?? | 0:00.06 | /System/Library/PrivateFrameworks/MobileDevice.framework/Versions/A/Resources/usbmuxd - launchd |
| 0 | 81 | 1 | 0 | 12:59AM | ?? | 0:00.48 | /usr/sbin/securityd -i |
| 205 | 83 | 1 | 0 | 12:59AM | ?? | 0:00.71 | /usr/libexec/locationd |
| 0 | 85 | 1 | 0 | 12:59AM | ?? | 0:00.25 | /usr/libexec/DuetHeuristic-BM |
| 244 | 86 | 1 | 0 | 12:59AM | ?? | 0:00.03 | /usr/libexec/displaypolicyd -k 1 |
| 0 | 87 | 1 | 0 | 12:59AM | ?? | 0:00.20 | /usr/sbin/blued |
| 0 | 88 | 1 | 0 | 12:59AM | ?? | 0:00.02 | autofs |
| 0 | 92 | 1 | 0 | 12:59AM | ?? | 0:00.08 | /System/Library/PrivateFrameworks/GenerationalStorage.framework/Versions/A/Support/revisi |
| 501 | 93 | 1 | 0 | 12:59AM | ?? | 0:00.69 | /System/Library/CoreServices/loginwindow.app/Contents/MacOS/loginwindow console |
| 0 | 94 | 1 | 0 | 12:59AM | ?? | 0:00.04 | /System/Library/CoreServices/login |
| 0 | 95 | 1 | 0 | 12:59AM | ?? | 0:00.01 | /usr/sbin/KernelEventAgent |
| 261 | 97 | 1 | 0 | 12:59AM | ?? | 0:04.27 | /usr/libexec/hidd |
| 0 | 98 | 1 | 0 | 12:59AM | ?? | 0:00.09 | /usr/libexec/AirPlayXPCHelper |
| 0 | 99 | 1 | 0 | 12:59AM | ?? | 0:00.75 | /usr/sbin/notifyd |

- a. Airportd** — the daemon for the wireless connectivity of the computer
- b. Warmd** — Runs once the boot happens and warms system wide resources. Certain resources which are frequently used by the user are pre warmed by the warmd process to reflect in the recent/frequent items list.
- c. Kextd** — They perform the load and unload of the kernel extensions such as device drivers as they are needed.
- d. Notifyd** — Forwards the Event Notifications between processes.
- e. Syslogd** — Performs Logging of the system status and Error messages

- Windows OS:

Task Manager

File Options View

Processes Performance App history Startup Users Details Services

| Name | 4% CPU | 77% Memory | 10% Disk | 0% Network |
|--|-----------|---------------|-------------|---------------|
| Google Chrome (32 bit) | 0% | 193.4 MB | 0 MB/s | 0 Mbps |
| > Antimalware Service Executable | 0% | 72.5 MB | 0 MB/s | 0 Mbps |
| Google Chrome (32 bit) | 0% | 63.1 MB | 0 MB/s | 0 Mbps |
| > Google Chrome (32 bit) | 0% | 59.9 MB | 0.1 MB/s | 0 Mbps |
| > Skype (32 bit) | 0% | 56.5 MB | 0 MB/s | 0 Mbps |
| > Microsoft Word (32 bit) | 0% | 54.3 MB | 0 MB/s | 0 Mbps |
| > Notepad++ : a free (GNU) source code editor (32 bit) | 0% | 46.1 MB | 0 MB/s | 0 Mbps |
| ByteFence Anti-Malware | 0% | 40.0 MB | 0 MB/s | 0 Mbps |
| Desktop Window Manager | 1.4% | 35.5 MB | 0 MB/s | 0 Mbps |
| Windows Explorer | 0.4% | 34.1 MB | 0 MB/s | 0 Mbps |
| > Microsoft Office Click-to-Run (SxS) | 0% | 27.9 MB | 0 MB/s | 0 Mbps |
| Windows Audio Device Graph Isolation | 0% | 25.7 MB | 0 MB/s | 0 Mbps |
| > Microsoft Windows Search Indexer | 0% | 22.7 MB | 0 MB/s | 0 Mbps |
| > Service Host: Local System (16) | 0% | 20.9 MB | 0 MB/s | 0 Mbps |
| Google Chrome (32 bit) | 0% | 20.6 MB | 0 MB/s | 0 Mbps |
| Google Chrome (32 bit) | 0% | 20.0 MB | 0 MB/s | 0 Mbps |
| Runtime Broker | 0% | 16.8 MB | 0 MB/s | 0 Mbps |
| > Service Host: Local Service (No Network) (4) | 0% | 16.4 MB | 0 MB/s | 0 Mbps |

a. Antimalware Service Executable:

It is a process of the Windows Defender which provides Real Time Protection by continuously monitoring the file system to take quick action against malwares. Hence often causing high memory usage.

b. Service Host: Local System:

It is a process that in turn hosts all the other services to perform various functions. It contains multiple sub processes to perform various sets of functions.

c. Runtime Broker:

It is a process that helps manage the permissions on the computer for applications from the Windows store. Hence, this process was introduced during Windows8 when the windows store was introduced. It helps managing the permissions of all the downloaded applications to access various features like location or microphone.

Record every computer you encounter for an entire day. Remember:

Laptop: Windows 10 , Ubuntu (Linux), MAC OS Sierra

PCs use Windows 10/8 or Ubuntu

Mac Book uses MacOS Sierra/Maverick

Server Machines use Windows Server / Linux CENTOS or Debian

SmartPhone: Android (Cynogen, MIUI), IOS

All the Smartphones and Ipad use Android OS or IOS respectively. Some phones have Cynogen or MIUI which are modified versions of Android

Tablet: Android, IOS

All the Samsung Tablets and Ipad use Android OS or IOS respectively

Watch: Google- Android Wear, Samsung- Tizen, Apple- watchOS

Websites you Visit: web servers are often hosted with linux(CentOS or RedHat or Debian) for most of the web sites.

Credit Card with Chip: For some of the smart cards, Chip Operating System (COS) is used which is also known as Mask - Masktech

Settop Box:

Embedded application Oss like OS-9, JavaOS, AperiOS or PowerTV

ATM Machine: Windows

Back in India, I have seen ATM machines booting up or crashing down with windows logo. The Machines must have been running an app on top of windows.

Routers and Switches: Network OS – Cisco IOS, Extreme Networks EXOS

Routers and Switches run Operating Systems with a Modified Linux Kernel that is used to work with underlying network hardware. Dedicated Network protocol applications are built on top of it.

Microwave:

It uses a combination of Embedded OS (Application Specific) - Electric Operating Systems, Computer OS.

Write a C program that performs memory allocation, write “HelloWorld” and free memory

XV6 Code:

```
$ srinivas@srinivas-Lenovo-Flex-2-14:~/Desktop/xv6 OS/xv6-public$ cat memoryalloc.c
```

```
#include "user.h"
#include "stat.h"
#include "types.h"
int main(void)
{
    int i;
    char *a;
    a = (char *) malloc(11);
    char *start;
    start = a;
    printf(1,"Start Address is : %d\n",start);
    char b[11] = "HelloWorld!";
    for(i = 0; b[i] != '\0'; i++)
    {
        *a = b[i];
        printf(1,"Next Address is : %d\n",a);
        a++;
    }
}
```

```

    }

    *a = '\0';
    for(a = start; *a != '\0'; a++) {
        printf(1,"The Data written is : %c at %d\n",*a,a);
    }
    a = start;
    printf(1,"The Data is : %s\n",a);
    free(a);
    printf(1,"The Data Fetch after Memory Freeing is : %s\n",*a);
    printf(1,"The Address after Memory Freeing is : %d\n",a);
    exit();
}
srinivas@srinivas-Lenovo-Flex-2-14:~/Desktop/xv6 OS/xv6-public$

```

Output:

```

rinivas@srinivas-Lenovo-Flex-2-14:~/Desktop/xv6 OS/xv6-public$ sudo make qemu
qemu-system-i386 -serial mon:stdio -drive file=fs.img,index=1,media=disk,format=raw -drive
file=xv6.img,index=0,media=disk,format=raw -smp 2 -m 512
xv6...
cpu1: starting
cpu0: starting
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap start 58
init: starting sh
$
$
$ ls
.          1 1 512
..         1 1 512
README    2 2 2487
cat        2 3 14340
echo       2 4 13205
forktest   2 5 8067
grep       2 6 15892
init       2 7 14090
kill       2 8 13245
ln         2 9 13171
ls         2 10 16015
mkdir      2 11 13266
rm         2 12 13247
sh         2 13 24671
stressfs   2 14 14137
usertests  2 15 67093

```

```

wc          2 16 15026
zombie      2 17 12915
hello       2 18 12930
memoryalloc 2 19 14124
console     3 20 0
$ memoryalloc
Start Address is : 45040
Next Address is : 45040
Next Address is : 45041
Next Address is : 45042
Next Address is : 45043
Next Address is : 45044
Next Address is : 45045
Next Address is : 45046
Next Address is : 45047
Next Address is : 45048
Next Address is : 45049
Next Address is : 45050
Next Address is : 45051
Next Address is : 45052
The Data written is : H at 45040
The Data written is : e at 45041
The Data written is : l at 4504
The Data written is : l at 45043
The Data written is : o at 45044
The Data written is : W at 45045
The Data written is : o at 45046
The Data written is : r at 45047
The Data written is : l at 45048
The Data written is : d at 45049
The Data written is : ! at 45050
The Data written is :  at 45051
The Data written is :  at 45052
The Data is : HelloWorld!
The Data Fetch after Memory Freeing is : D$Hell D$Wor D$!t&
The Address after Memory Freeing is : 45040
$

```

Linux Code:

```

srinivas@srinivas-Lenovo-Flex-2-14:~/Desktop$ cat mhello.c
#include "stdio.h"
#include "stdlib.h"
#include "string.h"
int main(void)
{

```

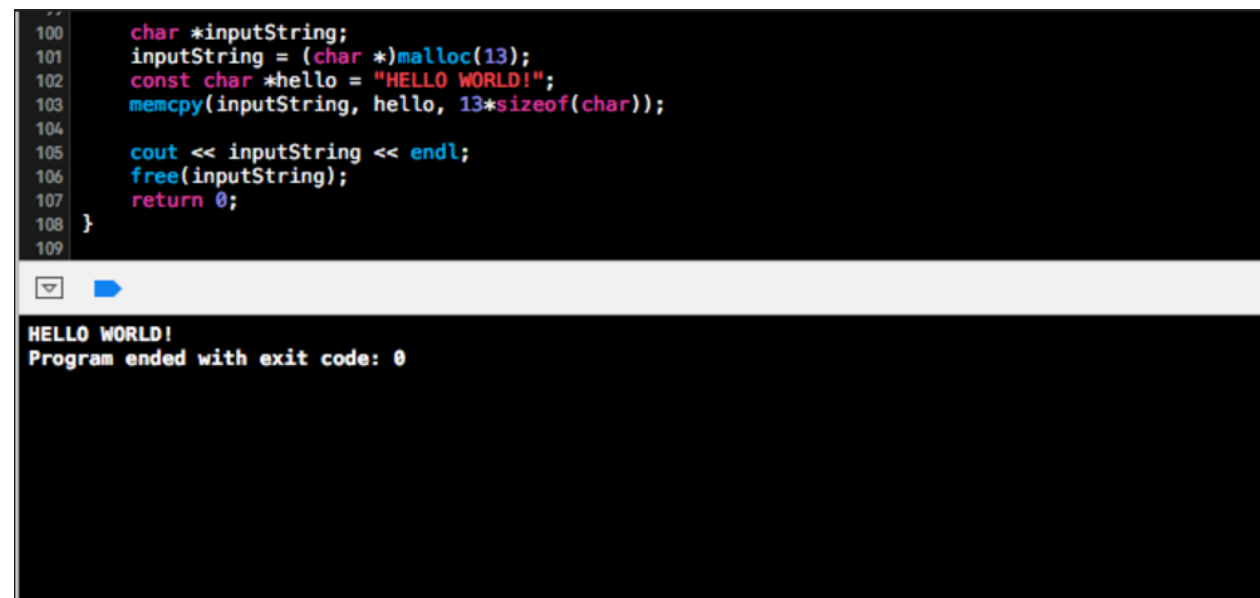
```
char *a;  
a = (char *) malloc(11);  
strcpy(a,"HelloWorld");  
printf("%s\n",a);  
free(a);  
printf("%s\n",a);  
}
```

Output:

```
srinivas@srinivas-Lenovo-Flex-2-14:~/Desktop$ gcc mhello.c -o mhello  
srinivas@srinivas-Lenovo-Flex-2-14:~/Desktop$ ./mhello  
HelloWorld
```

```
srinivas@srinivas-Lenovo-Flex-2-14:~/Desktop$
```

Another Implementation:



```
100     char *inputString;  
101     inputString = (char *)malloc(13);  
102     const char *hello = "HELLO WORLD!";  
103     memcpy(inputString, hello, 13*sizeof(char));  
104  
105     cout << inputString << endl;  
106     free(inputString);  
107     return 0;  
108 }  
109
```

HELLO WORLD!
Program ended with exit code: 0