Adam Prieto

Professor Ranjidha Rajan

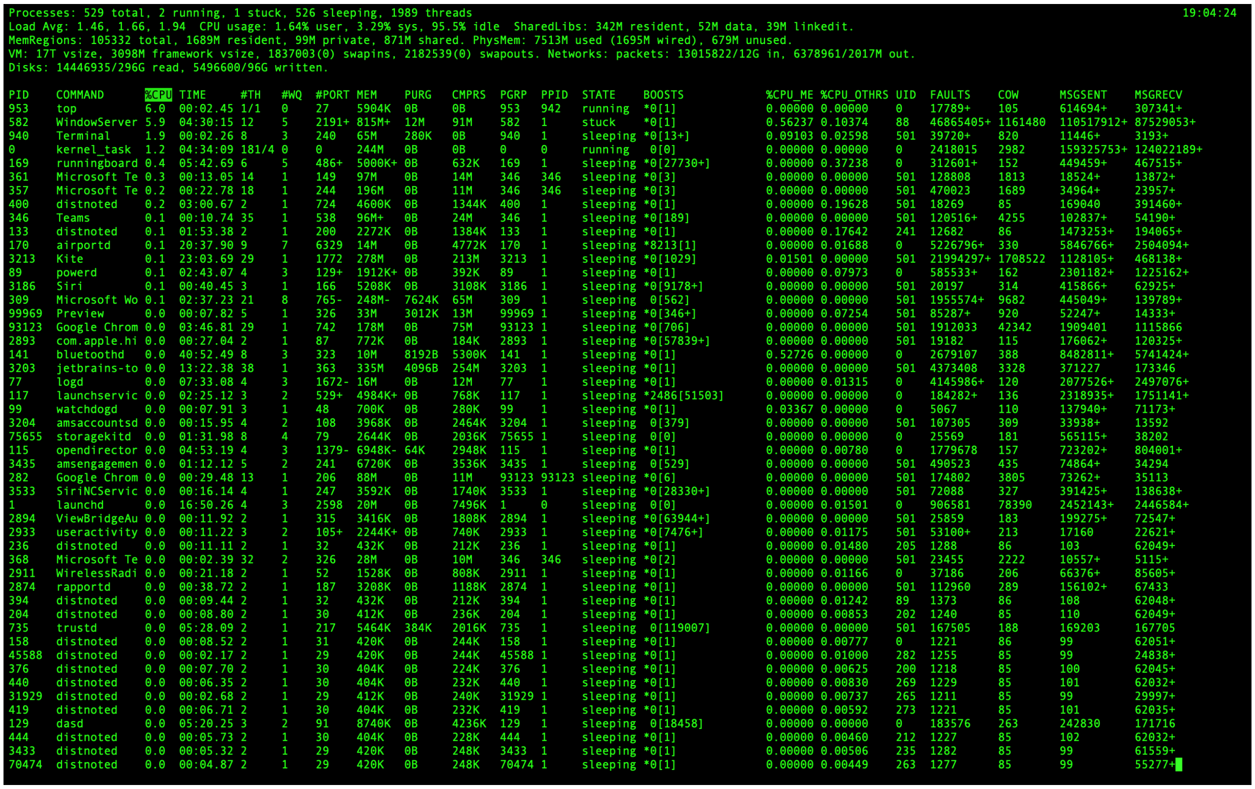
CS 3600 - Operating Systems

13 February 2022

Homework 1

1.) Indicate whether each series of state transitions for a process is valid or invalid. Justify your answer if its invalid.
    a. New -> ready -> blocked -> ready.
        i. This process is invalid since it's impossible for any series to go from ready to blocked. The correct process would be from ready to running to blocked.
    b. Running -> blocked -> ready -> blocked.
        i. Again, this process is invalid since you can't go from ready to blocked.
    c. New -> ready -> running -> ready.
        i. This process is valid.
2.) From the list, indicate which PCB fields will not change during a **process's lifetime**. Why?
    a. Child, Parent, CPU_state and Process_state
        i. Child stores the child processes created by the parent process and will change since a process will not have the same number of children at all times. The parent process, however, can create a child process whenever it wants to.
        ii. Parent stores the parent process that created the current process and since any process can have only one parent, the parent is the only field that remains constant to ensure reliability.
        iii. Although similar to process_state, CPU_State stores how much of the CPU is being used for the process and it will change with time.
        iv. Process_state in a PCB indicates the current state the process is in, like ready, suspended, running, stopped and so on. Since these states can change throughout any processes' life, process_state will also change.
3.) From the list, indicate which PCB fields may change while a process is in the **running state.** Why?
    a. Child, Parent, CPU_state and Process_state
        i. Again, since child is created by parents, it can change with what the parent requires.
        ii. Parent does not change since parent stores what the overarching process is.
        iii. Since CPU_state has a time limit, it does not change state.
        iv. Process_state does change since the PCB also changes.
4.) Learn the top command to display the resource utilization statistics of processes.
    a. Open a terminal and type the top command.

b. Start a browser and see the effect on the top display.
c. Press control-Z to stop.
d. Observe and write what you noticed. What are some of the parameters that you see?
e. A screenshot of my terminal can be found below.

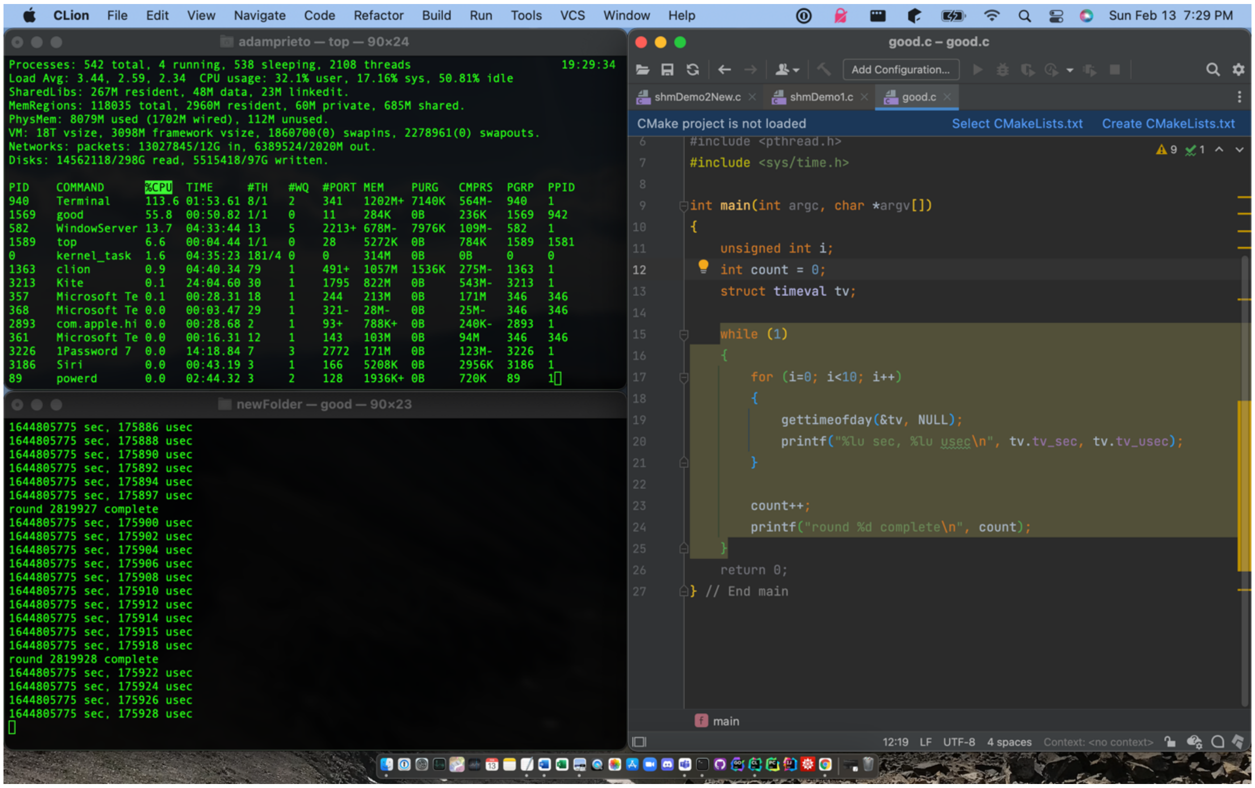


```
Processes: 529 total, 2 running, 1 stuck, 526 sleeping, 1989 threads                                                                  19:04:24
Load Avg: 1.46, 1.66, 1.94  CPU usage: 1.64% user, 3.29% sys, 95.5% idle  SharedLibs: 342M resident, 52M data, 39M linkedit.
MemRegions: 105332 total, 1689M resident, 99M private, 871M shared. PhysMem: 7513M used (1695M wired), 679M unused.
VM: 17T vsize, 3098M framework vsize, 1837003(0) swapins, 2182539(0) swapouts. Networks: packets: 13015822/12G in, 6378961/2017M out.
Disks: 14446935/296G read, 5496600/96G written.

PID    COMMAND      %CPU TIME     #TH    #WQ  #PORT MEM    PURG   CMPRS  PGRP  PPID  STATE    BOOSTS              %CPU_ME %CPU_OTHRS UID  FAULTS    COW     MSGSENT     MSGRECV
953    top          6.0  00:02.45 1/1    0    27    5904K  0B     0B     953   942   running  *0[1]               0.00000 0.00000    0    17789+    105     614694+     307341+
582    WindowServer 5.9  04:30:15 12     5    2191+ 815M+  12M    91M    582   1     stuck    *0[1]               0.56237 0.10374    88   46865405+ 1161480 110517912+  87529053+
940    Terminal     1.9  00:02.26 8      3    240   65M    280K   0B     940   1     sleeping *0[13+]             0.09103 0.02598    501  39720+    820     11446+      3193+
0      kernel_task  1.2  04:34:09 181/4  0    0     244M   0B     0B     0     0     running   0[0]               0.00000 0.00000    0    2418015   2982    159325753+  124022189+
169    runningboard 0.4  05:42.69 6      5    486+  5000K+ 0B     632K   169   1     sleeping *0[27730+]          0.00000 0.37238    0    312601+   152     449459+     467515+
361    Microsoft Te 0.3  00:13.05 14     1    149   97M    0B     14M    346   346   sleeping *0[3]               0.00000 0.00000    501  128808    1813    18524+      13872+
357    Microsoft Te 0.2  00:22.78 18     1    244   196M   0B     11M    346   346   sleeping *0[3]               0.00000 0.00000    501  470023    1689    34964+      23957+
400    distnoted    0.2  03:00.67 2      1    724   4600K  0B     1344K  400   1     sleeping *0[1]               0.00000 0.19628    501  18269     85      169040      391460+
346    Teams        0.1  00:10.74 35     1    538   96M+   0B     24M    346   1     sleeping *0[189]             0.00000 0.00000    501  120516+   4255    102837+     54190+
133    distnoted    0.1  01:53.38 2      1    200   2272K  0B     1384K  133   1     sleeping *0[1]               0.00000 0.17642    241  12682     86      1473253+    194065+
170    airportd     0.1  20:37.90 9      7    6329  14M    0B     4772K  170   1     sleeping *8213[1]            0.00000 0.01688    0    5226796+  330     5846766+    2504094+
3213   Kite         0.1  23:03.69 29     1    1772  278M   0B     213M   3213  1     sleeping *0[1029]            0.01501 0.00000    501  21994297+ 1708522 1128105+    468138+
89     powerd       0.1  02:43.07 4      3    129+  1912K+ 0B     392K   89    1     sleeping *0[1]               0.00000 0.07973    0    585533+   162     2301182+    1225162+
3186   Siri         0.1  00:40.45 3      1    166   5208K  0B     3108K  3186  1     sleeping *0[9178+]           0.00000 0.00000    501  20197     314     415866+     62925+
309    Microsoft Wo 0.1  02:37.23 21     8    765-  248M-  7624K  65M    309   1     sleeping  0[562]             0.00000 0.00000    501  1955574+  9682    445049+     139789+
99969  Preview      0.0  00:07.82 5      1    326   33M    3012K  13M    99969 1     sleeping *0[346+]            0.00000 0.07254    501  85287+    920     52247+      14333+
93123  Google Chrom 0.0  03:46.81 29     1    742   178M   0B     75M    93123 1     sleeping *0[706]             0.00000 0.00000    501  1912033   42342   1909401     1115866
2893   com.apple.hi 0.0  00:27.04 2      1    87    772K   0B     184K   2893  1     sleeping *0[57839+]          0.00000 0.00000    501  19182     115     176062+     120325+
141    bluetoothd   0.0  40:52.49 8      3    323   10M    8192B  5300K  141   1     sleeping *0[1]               0.52726 0.00000    0    2679107   388     8482811+    5741424+
3203   jetbrains-to 0.0  13:22.38 38     1    363   335M   4096B  254M   3203  1     sleeping *0[1]               0.00000 0.00000    501  4373408   3328    371227      173346
77     logd         0.0  07:33.08 4      3    1672- 16M    0B     12M    77    1     sleeping *0[1]               0.00000 0.01315    0    4145986+  120     2077526+    2497076+
117    launchservic 0.0  02:25.12 3      2    529+  4984K+ 0B     768K   117   1     sleeping *2486[51503]        0.00000 0.00000    0    184282+   136     2318935+    1751141+
99     watchdogd    0.0  00:07.91 3      1    48    700K   0B     280K   99    1     sleeping *0[1]               0.03367 0.00000    0    5067      110     137940+     71173+
3204   amsaccountsd 0.0  00:15.95 4      2    108   3968K  0B     2464K  3204  1     sleeping  0[379]             0.00000 0.00000    501  107305    309     33938+      13592
75655  storagekitd  0.0  01:31.98 8      4    79    2644K  0B     2036K  75655 1     sleeping  0[0]               0.00000 0.00000    0    25569     181     565115+     38202
115    opendirector 0.0  04:53.19 4      3    1379- 6948K- 64K    2948K  115   1     sleeping *0[1]               0.00000 0.00780    0    1779678   157     723202+     804001+
3435   amsengagemen 0.0  01:12.12 5      2    241   6720K  0B     3536K  3435  1     sleeping  0[529]             0.00000 0.00000    501  490523    435     74864+      34294
282    Google Chrom 0.0  00:29.48 13     1    206   88M    0B     11M    93123 93123 sleeping *0[6]               0.00000 0.00000    501  174802    3805    73262+      35113
3533   SiriNCServic 0.0  00:16.14 4      1    247   3592K  0B     1740K  3533  1     sleeping *0[28330+]          0.00000 0.00000    501  72088     327     391425+     138638+
1      launchd      0.0  16:50.26 4      3    2598  20M    0B     7496K  1     0     sleeping  0[0]               0.00000 0.01501    0    906581    78390   2452143+    2446584+
2894   ViewBridgeAu 0.0  00:11.92 2      1    315   3416K  0B     1808K  2894  1     sleeping *0[63944+]          0.00000 0.00000    501  25859     183     199275+     72547+
2933   useractivity 0.0  00:11.22 3      2    105+  2244K+ 0B     740K   2933  1     sleeping *0[7476+]           0.00000 0.01175    501  53100+    213     17160       22621+
236    distnoted    0.0  00:11.11 2      1    32    432K   0B     212K   236   1     sleeping *0[1]               0.00000 0.01480    205  1288      86      103         62049+
368    Microsoft Te 0.0  00:02.39 32     2    326   28M    0B     10M    346   346   sleeping *0[2]               0.00000 0.00000    501  23455     2222    10557+      5115+
2911   WirelessRadi 0.0  00:21.18 2      1    52    1528K  0B     808K   2911  1     sleeping *0[1]               0.00000 0.01166    0    37186     206     66376+      85605+
2874   rapportd     0.0  00:38.72 2      1    187   3208K  0B     1188K  2874  1     sleeping *0[1]               0.00000 0.00000    501  112960    289     156102+     67433
394    distnoted    0.0  00:09.44 2      1    32    432K   0B     212K   394   1     sleeping *0[1]               0.00000 0.01242    89   1373      86      108         62048+
204    distnoted    0.0  00:08.80 2      1    30    412K   0B     236K   204   1     sleeping *0[1]               0.00000 0.00853    202  1240      85      110         62049+
735    trustd       0.0  05:28.09 2      1    217   5464K  384K   2016K  735   1     sleeping  0[119007]          0.00000 0.00000    501  167505    188     169203      167705
158    distnoted    0.0  00:08.52 2      1    31    420K   0B     244K   158   1     sleeping *0[1]               0.00000 0.00777    0    1221      86      99          62051+
45588  distnoted    0.0  00:02.17 2      1    29    420K   0B     244K   45588 1     sleeping *0[1]               0.00000 0.01000    282  1255      85      99          24838+
376    distnoted    0.0  00:07.70 2      1    30    404K   0B     224K   376   1     sleeping *0[1]               0.00000 0.00625    200  1218      85      100         62045+
440    distnoted    0.0  00:06.35 2      1    30    404K   0B     232K   440   1     sleeping *0[1]               0.00000 0.00830    269  1229      85      101         62032+
31929  distnoted    0.0  00:02.68 2      1    29    412K   0B     240K   31929 1     sleeping *0[1]               0.00000 0.00737    265  1211      85      99          29997+
419    distnoted    0.0  00:06.71 2      1    30    404K   0B     232K   419   1     sleeping *0[1]               0.00000 0.00592    273  1221      85      101         62035+
129    dasd         0.0  05:20.25 3      2    91    8740K  0B     4236K  129   1     sleeping  0[18458]           0.00000 0.00000    0    183576    263     242830      171716
444    distnoted    0.0  00:05.73 2      1    30    404K   0B     228K   444   1     sleeping *0[1]               0.00000 0.00460    212  1227      85      102         62032+
3433   distnoted    0.0  00:05.32 2      1    29    420K   0B     248K   3433  1     sleeping *0[1]               0.00000 0.00506    235  1282      85      99          61559+
70474  distnoted    0.0  00:04.87 2      1    29    420K   0B     248K   70474 1     sleeping *0[1]               0.00000 0.00449    263  1277      85      99          55277+
```

f. From the above screenshot, I can see that there are several common programs running on my computer since I had to write this very document in Microsoft word and also had some other programs running in the background. Despite the fact that I had several processes running on my computer, my CPU did not see maximum utilization. The top terminal also listed several other parameters including memory usage, port number, and others.

4.2) Write the C program, program 2, let the top command run in the old terminal and compile the program in a new terminal and observe the parameters of top command, observe which process is taking more cpu? Which process has more memory share?

a. Below is a screenshot of my terminal running top, a screenshot of my terminal running the program, and a screenshot of the code in my IDE. As you can see, the terminal command is using the most amount of CPU and memory at 113.6% and >1202 MB respectively. Other resource heavy operations include the second terminal window and the IDE itself. With all of this included, it becomes pretty apparent that the program was designed to maximize resource usage and see the results.



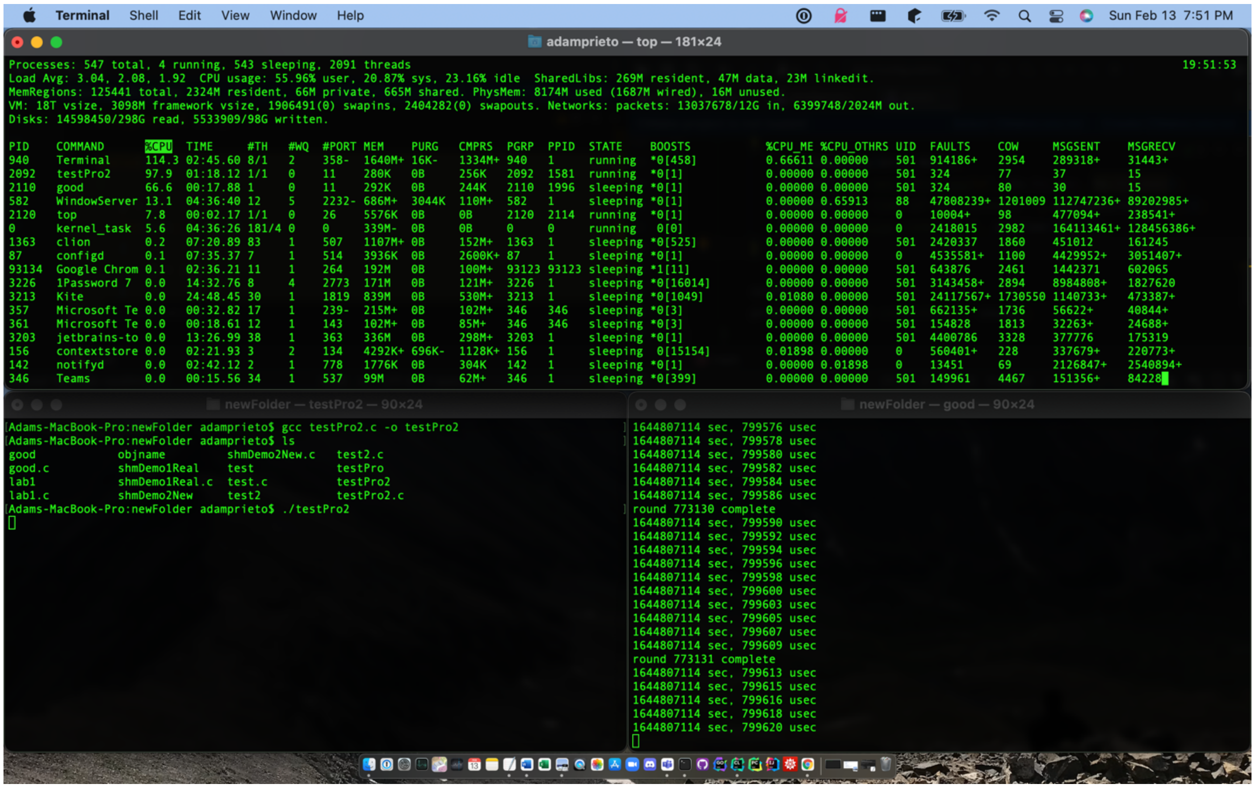
```
Processes: 542 total, 4 running, 538 sleeping, 2108 threads                     19:29:34
Load Avg: 3.44, 2.59, 2.34  CPU usage: 32.1% user, 17.16% sys, 50.81% idle
SharedLibs: 267M resident, 48M data, 23M linkedit.
MemRegions: 118035 total, 2960M resident, 60M private, 685M shared.
PhysMem: 8079M used (1702M wired), 112M unused.
VM: 18T vsize, 3098M framework vsize, 1860700(0) swapins, 2278961(0) swapouts.
Networks: packets: 13027845/12G in, 6389524/2020M out.
Disks: 14562118/298G read, 5515418/97G written.

PID    COMMAND      %CPU  TIME     #TH   #WQ  #PORT MEM    PURG   CMPRS  PGRP PPID
940    Terminal     113.6 01:53.61 8/1   2    341   1202M+ 7140K  564M-  940  1
1569   good         55.8  00:50.82 1/1   0    11    284K   0B     236K   1569 942
582    WindowServer 13.7  04:33:44 13    5    2213+ 678M-  7976K  109M-  582  1
1589   top          6.6   00:04.44 1/1   0    28    5272K  0B     784K   1589 1581
0      kernel_task  1.6   04:35:23 181/4 0    0     314M   0B     0B     0    0
1363   clion        0.9   04:40.34 79    1    491+  1057M  1536K  275M-  1363 1
3213   Kite         0.1   24:04.60 30    1    1795  822M   0B     543M-  3213 1
357    Microsoft Te 0.1   00:28.31 18    1    244   213M   0B     171M   346  346
368    Microsoft Te 0.0   00:03.47 29    1    321-  28M-   0B     25M-   346  346
2893   com.apple.hi 0.0   00:28.68 2     1    93+   788K+  0B     240K-  2893 1
361    Microsoft Te 0.0   00:16.31 12    1    143   103M   0B     94M    346  346
3226   1Password 7  0.0   14:18.84 7     3    2772  171M   0B     123M-  3226 1
3186   Siri         0.0   00:43.19 3     1    166   5208K  0B     2956K  3186 1
89     powerd       0.0   02:44.32 3     2    128   1936K+ 0B     720K   89   1
```

```
1644805775 sec, 175886 usec
1644805775 sec, 175888 usec
1644805775 sec, 175890 usec
1644805775 sec, 175892 usec
1644805775 sec, 175894 usec
1644805775 sec, 175897 usec
round 2819927 complete
1644805775 sec, 175900 usec
1644805775 sec, 175902 usec
1644805775 sec, 175904 usec
1644805775 sec, 175906 usec
1644805775 sec, 175908 usec
1644805775 sec, 175910 usec
1644805775 sec, 175912 usec
1644805775 sec, 175914 usec
1644805775 sec, 175915 usec
1644805775 sec, 175918 usec
round 2819928 complete
1644805775 sec, 175922 usec
1644805775 sec, 175924 usec
1644805775 sec, 175926 usec
1644805775 sec, 175928 usec
```

```c
#include <pthread.h>
#include <sys/time.h>

int main(int argc, char *argv[])
{
    unsigned int i;
    int count = 0;
    struct timeval tv;

    while (1)
    {
        for (i=0; i<10; i++)
        {
            gettimeofday(&tv, NULL);
            printf("%lu sec, %lu usec\n", tv.tv_sec, tv.tv_usec);
        }

        count++;
        printf("round %d complete\n", count);
    }
    return 0;
} // End main
```

4.3) Write and compile both program 1 and program 2 in separate files IDE files and then execute thee results in separate terminals. Compare the effect of their cpu share using the top display in a third terminal and write any observations.

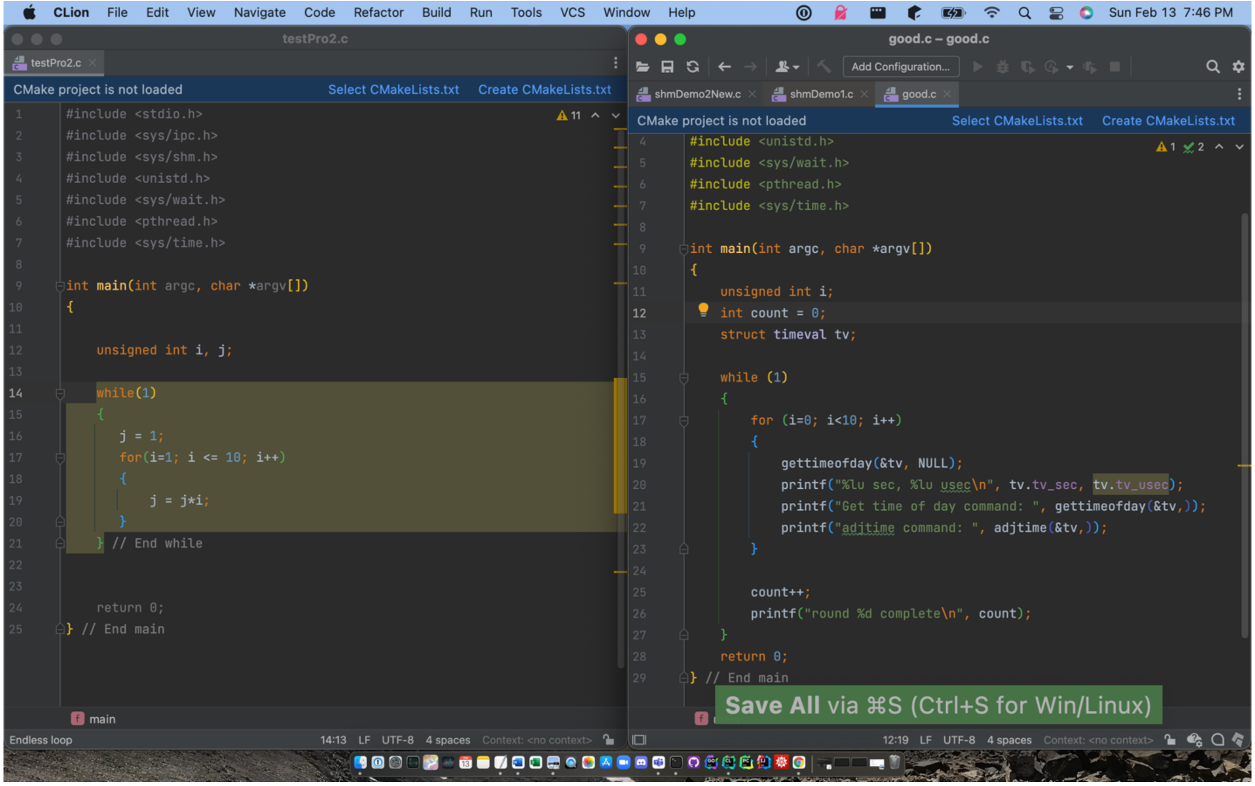
a. Relevant screenshots are again provided below. As you can see, the cpu is again performing many operations with large chunks of memory also taken up by the various processes. Since these processes are using so many resources on my mac, it becomes even more apparent that this problems was also designed to see what happens when you have several programs running on your computer at once.



```
Processes: 547 total, 4 running, 543 sleeping, 2091 threads                                                                                                              19:51:53
Load Avg: 3.04, 2.08, 1.92  CPU usage: 55.96% user, 20.87% sys, 23.16% idle  SharedLibs: 269M resident, 47M data, 23M linkedit.
MemRegions: 125441 total, 2324M resident, 66M private, 665M shared. PhysMem: 8174M used (1687M wired), 16M unused.
VM: 18T vsize, 3098M framework vsize, 1906491(0) swapins, 2404282(0) swapouts. Networks: packets: 13037678/12G in, 6399748/2024M out.
Disks: 14598450/298G read, 5533909/98G written.

PID    COMMAND      %CPU  TIME     #TH    #WQ  #PORT MEM    PURG   CMPRS  PGRP  PPID  STATE    BOOSTS          %CPU_ME %CPU_OTHRS UID  FAULTS    COW     MSGSENT    MSGRECV
940    Terminal     114.3 02:45.60 8/1    2    358-  1640M+ 16K-   1334M+ 940   1     running  *0[458]         0.66611 0.00000    501  914186+   2954    289318+    31443+
2092   testPro2     97.9  01:18.12 1/1    0    11    280K   0B     256K   2092  1581  running  *0[1]           0.00000 0.00000    501  324       77      37         15
2110   good         66.6  00:17.88 1      0    11    292K   0B     244K   2110  1996  sleeping *0[1]           0.00000 0.00000    501  324       80      30         15
582    WindowServer 13.1  04:36:40 12     5    2232- 686M+  3044K  110M+  582   1     sleeping *0[1]           0.00000 0.65913    88   47808239+ 1201009 112747236+ 89202985+
2120   top          7.8   00:02.17 1/1    0    26    5576K  0B     0B     2120  2114  running  *0[1]           0.00000 0.00000    0    10004+    98      477094+    238541+
0      kernel_task  5.6   04:36:26 181/4  0    0     339M-  0B     0B     0     0     running   0[0]           0.00000 0.00000    0    2418015   2982    164113461+ 128456386+
1363   clion        0.2   07:20.89 83     1    507   1107M+ 0B     152M+  1363  1     sleeping *0[525]         0.00000 0.00000    501  2420337   1860    451012     161245
87     configd      0.1   07:35.37 7      1    514   3936K  0B     2600K+ 87    1     sleeping *0[1]           0.00000 0.00000    0    4535581+  1100    4429952+   3051407+
93134  Google Chrom 0.1   02:36.21 11     1    264   192M   0B     100M+  93123 93123 sleeping *1[11]          0.00000 0.00000    501  643876    2461    1442371    602065
3226   1Password 7  0.0   14:32.76 8      4    2773  171M   0B     121M+  3226  1     sleeping *0[16014]       0.00000 0.00000    501  3143458+  2894    8984808+   1827620
3213   Kite         0.0   24:48.45 30     1    1819  839M   0B     530M+  3213  1     sleeping *0[1049]        0.01080 0.00000    501  24117567+ 1730550 1140733+   473387+
357    Microsoft Te 0.0   00:32.82 17     1    239-  215M+  0B     102M+  346   346   sleeping *0[3]           0.00000 0.00000    501  662135+   1736    56622+     40844+
361    Microsoft Te 0.0   00:18.61 12     1    143   102M+  0B     85M+   346   346   sleeping *0[3]           0.00000 0.00000    501  154828    1813    32263+     24688+
3203   jetbrains-to 0.0   13:26.99 38     1    363   336M   0B     298M+  3203  1     sleeping *0[1]           0.00000 0.00000    501  4400786   3328    377776     175319
156    contextstore 0.0   02:21.93 3      2    134   4292K+ 696K-  1128K+ 156   1     sleeping  0[15154]       0.01898 0.00000    0    560401+   228     337679+    220773+
142    notifyd      0.0   02:42.12 2      1    778   1776K  0B     304K   142   1     sleeping *0[1]           0.00000 0.01898    0    13451     69      2126847+   2540894+
346    Teams        0.0   00:15.56 34     1    537   99M    0B     62M+   346   1     sleeping *0[399]         0.00000 0.00000    501  149961    4467    151356+    84228
```

```
Adams-MacBook-Pro:newFolder adamprieto$ gcc testPro2.c -o testPro2
Adams-MacBook-Pro:newFolder adamprieto$ ls
good            objname         shmDemo2New.c   test2.c
good.c          shmDemo1Real    test            testPro
lab1            shmDemo1Real.c  test.c          testPro2
lab1.c          shmDemo2New     test2           testPro2.c
Adams-MacBook-Pro:newFolder adamprieto$ ./testPro2
```

```
1644807114 sec, 799576 usec
1644807114 sec, 799578 usec
1644807114 sec, 799580 usec
1644807114 sec, 799582 usec
1644807114 sec, 799584 usec
1644807114 sec, 799586 usec
round 773130 complete
1644807114 sec, 799590 usec
1644807114 sec, 799592 usec
1644807114 sec, 799594 usec
1644807114 sec, 799596 usec
1644807114 sec, 799598 usec
1644807114 sec, 799600 usec
1644807114 sec, 799603 usec
1644807114 sec, 799605 usec
1644807114 sec, 799607 usec
1644807114 sec, 799609 usec
round 773131 complete
1644807114 sec, 799613 usec
1644807114 sec, 799615 usec
1644807114 sec, 799616 usec
1644807114 sec, 799618 usec
1644807114 sec, 799620 usec
```



```
#include <stdio.h>
#include <sys/ipc.h>
#include <sys/shm.h>
#include <unistd.h>
#include <sys/wait.h>
#include <pthread.h>
#include <sys/time.h>

int main(int argc, char *argv[])
{

    unsigned int i, j;

    while(1)
    {
        j = 1;
        for(i=1; i <= 10; i++)
        {
            j = j*i;
        }
    } // End while


    return 0;
} // End main
```

```
#include <unistd.h>
#include <sys/wait.h>
#include <pthread.h>
#include <sys/time.h>

int main(int argc, char *argv[])
{
    unsigned int i;
    int count = 0;
    struct timeval tv;

    while (1)
    {
        for (i=0; i<10; i++)
        {
            gettimeofday(&tv, NULL);
            printf("%lu sec, %lu usec\n", tv.tv_sec, tv.tv_usec);
            printf("Get time of day command: ", gettimeofday(&tv,));
            printf("adjtime command: ", adjtime(&tv,));
        }

        count++;
        printf("round %d complete\n", count);
    }
    return 0;
} // End main
```