check process + process states

CSE3241: Operating System and System Programming

Class-3

Sangeeta Biswas, Ph.D.
Assistant Professor
Dept. of Computer Science and Engineering (CSE)
Faculty of Engineering
University of Rajshahi (RU)
Rajshahi-6205, Bangladesh
E-mail: sangeeta.cse@ru.ac.bd

Concepts of Process

1. Process is a program in execution.
2. Program is a passive entity stored on disk (executable file); process is an active entity.
3. Program becomes process when an executable file is loaded into memory.
4. One program can have several processes
   - multiple users or one user can execute multiple copies of the same program at a time.
5. Process execution must progress in sequential fashion.
   - No parallel execution of instructions of a single process.
6. Modern OS manages thousand of processes of a variety of application software, system software as well as its own processes at a time.

Check Process in Linux based OS

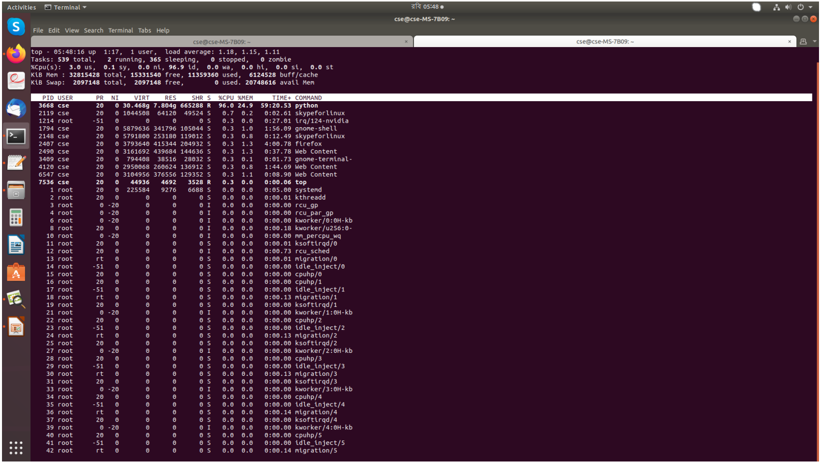
To see all processes, currently managed by a Linux based OS (e.g., Ubuntu), at real time: win + R => type cmd => enter

- Open a terminal, the black screen, by typing Ctrl + Alt + T .
- Type 'top' and press Enter. Press 'Q' to exit. tasklist
- $ top

To get a snapshot of all processes, currently managed by a Linux based OS (e.g., Ubuntu): $ ps aux

Check Process in Linux based OS

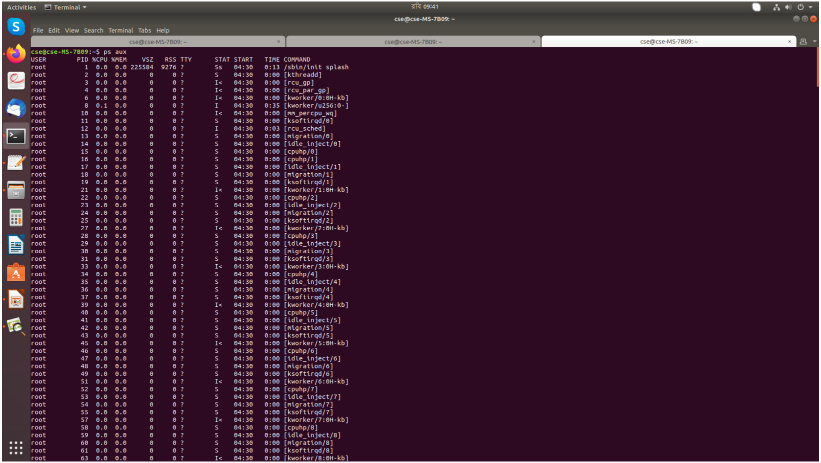
$ top



```
top - 05:48:16 up  1:17,  1 user,  load average: 1.18, 1.15, 1.11
Tasks: 539 total,   2 running, 365 sleeping,   0 stopped,   0 zombie
%Cpu(s):  3.0 us,  0.1 sy,  0.0 ni, 96.9 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem : 32815428 total, 15331540 free, 11359360 used,  6124528 buff/cache
KiB Swap:  2097148 total,  2097148 free,        0 used. 20748616 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
 3668 cse       20   0 30.468g 7.804g 665288 R  96.0 24.9  59:20.53 python
 2119 cse       20   0 1044508  64120  49524 S   0.7  0.2   0:02.61 skypeforlinux
 1214 root     -51   0       0      0      0 S   0.3  0.0   0:27.01 irq/124-nvidia
 1794 cse       20   0 5879636 341796 105044 S   0.3  1.0   1:56.09 gnome-shell
 2148 cse       20   0 5791800 253180 119012 S   0.3  0.8   0:12.49 skypeforlinux
 2407 cse       20   0 3793640 415344 204932 S   0.3  1.3   4:00.78 firefox
 2490 cse       20   0 3161692 439684 144636 S   0.3  1.3   0:37.78 Web Content
 3409 cse       20   0  794408  38516  28032 S   0.3  0.1   0:01.73 gnome-terminal-
 4120 cse       20   0 2950068 260624 136912 S   0.3  0.8   1:44.69 Web Content
 6547 cse       20   0 3104956 376556 129352 S   0.3  1.1   0:08.90 Web Content
 7536 cse       20   0   44936   4692   3528 R   0.3  0.0   0:00.06 top
    1 root      20   0  225584   9276   6688 S   0.0  0.0   0:05.00 systemd
    2 root      20   0       0      0      0 S   0.0  0.0   0:00.01 kthreadd
    3 root       0 -20       0      0      0 I   0.0  0.0   0:00.00 rcu_gp
    4 root       0 -20       0      0      0 I   0.0  0.0   0:00.00 rcu_par_gp
    6 root       0 -20       0      0      0 I   0.0  0.0   0:00.00 kworker/0:0H-kb
    8 root      20   0       0      0      0 I   0.0  0.0   0:00.18 kworker/u256:0-
   10 root       0 -20       0      0      0 I   0.0  0.0   0:00.00 mm_percpu_wq
   11 root      20   0       0      0      0 S   0.0  0.0   0:00.01 ksoftirqd/0
   12 root      20   0       0      0      0 I   0.0  0.0   0:00.73 rcu_sched
   13 root      rt   0       0      0      0 S   0.0  0.0   0:00.01 migration/0
   14 root     -51   0       0      0      0 S   0.0  0.0   0:00.00 idle_inject/0
   15 root      20   0       0      0      0 S   0.0  0.0   0:00.00 cpuhp/0
   16 root      20   0       0      0      0 S   0.0  0.0   0:00.00 cpuhp/1
   17 root     -51   0       0      0      0 S   0.0  0.0   0:00.00 idle_inject/1
   18 root      rt   0       0      0      0 S   0.0  0.0   0:00.13 migration/1
   19 root      20   0       0      0      0 S   0.0  0.0   0:00.00 ksoftirqd/1
   21 root       0 -20       0      0      0 I   0.0  0.0   0:00.00 kworker/1:0H-kb
   22 root      20   0       0      0      0 S   0.0  0.0   0:00.00 cpuhp/2
   23 root     -51   0       0      0      0 S   0.0  0.0   0:00.00 idle_inject/2
   24 root      rt   0       0      0      0 S   0.0  0.0   0:00.13 migration/2
   25 root      20   0       0      0      0 S   0.0  0.0   0:00.00 ksoftirqd/2
   27 root       0 -20       0      0      0 I   0.0  0.0   0:00.00 kworker/2:0H-kb
   28 root      20   0       0      0      0 S   0.0  0.0   0:00.00 cpuhp/3
   29 root     -51   0       0      0      0 S   0.0  0.0   0:00.00 idle_inject/3
   30 root      rt   0       0      0      0 S   0.0  0.0   0:00.13 migration/3
   31 root      20   0       0      0      0 S   0.0  0.0   0:00.00 ksoftirqd/3
   33 root       0 -20       0      0      0 I   0.0  0.0   0:00.00 kworker/3:0H-kb
   34 root      20   0       0      0      0 S   0.0  0.0   0:00.00 cpuhp/4
   35 root     -51   0       0      0      0 S   0.0  0.0   0:00.00 idle_inject/4
   36 root      rt   0       0      0      0 S   0.0  0.0   0:00.14 migration/4
   37 root      20   0       0      0      0 S   0.0  0.0   0:00.00 ksoftirqd/4
   39 root       0 -20       0      0      0 I   0.0  0.0   0:00.00 kworker/4:0H-kb
   40 root      20   0       0      0      0 S   0.0  0.0   0:00.00 cpuhp/5
   41 root     -51   0       0      0      0 S   0.0  0.0   0:00.00 idle_inject/5
   42 root      rt   0       0      0      0 S   0.0  0.0   0:00.14 migration/5
```

Check Process in Linux based OS

$ ps aux

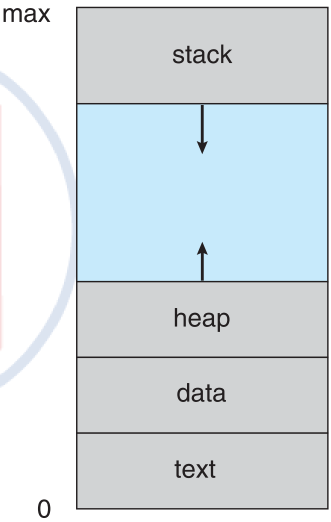


```
cse@cse-MS-7B09:~$ ps aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.0  0.0 225584  9276 ?        Ss   04:30   0:13 /sbin/init splash
root           2  0.0  0.0      0     0 ?        S    04:30   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        I<   04:30   0:00 [rcu_gp]
root           4  0.0  0.0      0     0 ?        I<   04:30   0:00 [rcu_par_gp]
root           6  0.0  0.0      0     0 ?        I<   04:30   0:00 [kworker/0:0H-kb]
root           8  0.1  0.0      0     0 ?        I    04:30   0:35 [kworker/u256:0-]
root          10  0.0  0.0      0     0 ?        I<   04:30   0:00 [mm_percpu_wq]
root          11  0.0  0.0      0     0 ?        S    04:30   0:00 [ksoftirqd/0]
root          12  0.0  0.0      0     0 ?        I    04:30   0:03 [rcu_sched]
root          13  0.0  0.0      0     0 ?        S    04:30   0:00 [migration/0]
root          14  0.0  0.0      0     0 ?        S    04:30   0:00 [idle_inject/0]
root          15  0.0  0.0      0     0 ?        S    04:30   0:00 [cpuhp/0]
root          16  0.0  0.0      0     0 ?        S    04:30   0:00 [cpuhp/1]
root          17  0.0  0.0      0     0 ?        S    04:30   0:00 [idle_inject/1]
root          18  0.0  0.0      0     0 ?        S    04:30   0:00 [migration/1]
root          19  0.0  0.0      0     0 ?        S    04:30   0:00 [ksoftirqd/1]
root          21  0.0  0.0      0     0 ?        I<   04:30   0:00 [kworker/1:0H-kb]
root          22  0.0  0.0      0     0 ?        S    04:30   0:00 [cpuhp/2]
root          23  0.0  0.0      0     0 ?        S    04:30   0:00 [idle_inject/2]
root          24  0.0  0.0      0     0 ?        S    04:30   0:00 [migration/2]
root          25  0.0  0.0      0     0 ?        S    04:30   0:00 [ksoftirqd/2]
root          27  0.0  0.0      0     0 ?        I<   04:30   0:00 [kworker/2:0H-kb]
root          28  0.0  0.0      0     0 ?        S    04:30   0:00 [cpuhp/3]
root          29  0.0  0.0      0     0 ?        S    04:30   0:00 [idle_inject/3]
root          30  0.0  0.0      0     0 ?        S    04:30   0:00 [migration/3]
root          31  0.0  0.0      0     0 ?        S    04:30   0:00 [ksoftirqd/3]
root          33  0.0  0.0      0     0 ?        I<   04:30   0:00 [kworker/3:0H-kb]
root          34  0.0  0.0      0     0 ?        S    04:30   0:00 [cpuhp/4]
root          35  0.0  0.0      0     0 ?        S    04:30   0:00 [idle_inject/4]
root          36  0.0  0.0      0     0 ?        S    04:30   0:00 [migration/4]
root          37  0.0  0.0      0     0 ?        S    04:30   0:00 [ksoftirqd/4]
root          39  0.0  0.0      0     0 ?        I<   04:30   0:00 [kworker/4:0H-kb]
root          40  0.0  0.0      0     0 ?        S    04:30   0:00 [cpuhp/5]
root          41  0.0  0.0      0     0 ?        S    04:30   0:00 [idle_inject/5]
root          42  0.0  0.0      0     0 ?        S    04:30   0:00 [migration/5]
root          43  0.0  0.0      0     0 ?        S    04:30   0:00 [ksoftirqd/5]
root          45  0.0  0.0      0     0 ?        I<   04:30   0:00 [kworker/5:0H-kb]
root          46  0.0  0.0      0     0 ?        S    04:30   0:00 [cpuhp/6]
root          47  0.0  0.0      0     0 ?        S    04:30   0:00 [idle_inject/6]
root          48  0.0  0.0      0     0 ?        S    04:30   0:00 [migration/6]
root          49  0.0  0.0      0     0 ?        S    04:30   0:00 [ksoftirqd/6]
root          51  0.0  0.0      0     0 ?        I<   04:30   0:00 [kworker/6:0H-kb]
root          52  0.0  0.0      0     0 ?        S    04:30   0:00 [cpuhp/7]
root          53  0.0  0.0      0     0 ?        S    04:30   0:00 [idle_inject/7]
root          54  0.0  0.0      0     0 ?        S    04:30   0:00 [migration/7]
root          55  0.0  0.0      0     0 ?        S    04:30   0:00 [ksoftirqd/7]
root          57  0.0  0.0      0     0 ?        I<   04:30   0:00 [kworker/7:0H-kb]
root          58  0.0  0.0      0     0 ?        S    04:30   0:00 [cpuhp/8]
root          59  0.0  0.0      0     0 ?        S    04:30   0:00 [idle_inject/8]
root          60  0.0  0.0      0     0 ?        S    04:30   0:00 [migration/8]
root          61  0.0  0.0      0     0 ?        S    04:30   0:00 [ksoftirqd/8]
root          63  0.0  0.0      0     0 ?        I<   04:30   0:00 [kworker/8:0H-kb]
```

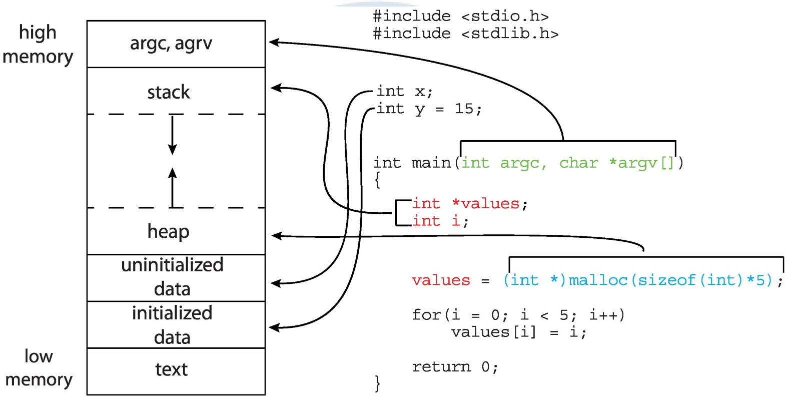
Process in Memory

A process has multiple parts, when it is in memory:

- **Text**: contains program code
- **Stack**: contains temporary data
  oitemsepFunction parameters, return addresses, local variables
- **Data**: contains global variables
- **Heap**: contains dynamically allocated memory during run time



Memory Layout of a C Program



high memory

argc, agrv

stack

heap

uninitialized data

initialized data

text

low memory

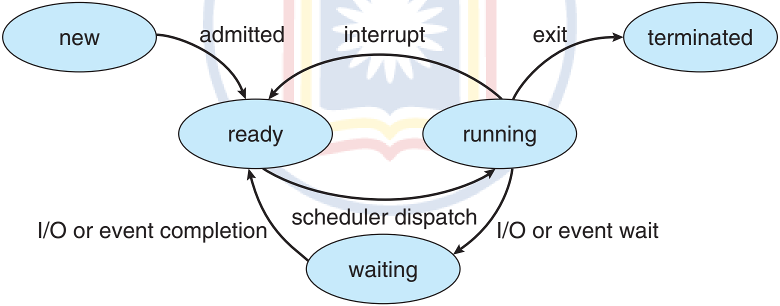
```
#include <stdio.h>
#include <stdlib.h>

int x;
int y = 15;

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

    values = (int *)malloc(sizeof(int)*5);

    for(i = 0; i < 5; i++)
        values[i] = i;

    return 0;
}
```

Process States

book 107

As a process executes, it changes state

- **New**: The process is being created
- **Running**: Instructions are being executed
- **Waiting**: The process is waiting for some event to occur
- **Ready**: The process is waiting to be assigned to a processor
- **Terminated**: The process has finished execution



Process State Code in Ubuntu

As a process executes, it changes state

- D : uninterruptible sleep (usually IO)
- R : running or runnable (on run queue)
- S : interruptible sleep (waiting for an event to complete)
- T : stopped by job control signal
- t : stopped by debugger during the tracing
- W : paging (not valid since the 2.6.xx kernel)
- X : dead (should never be seen)
- Z : defunct (”zombie”) process, terminated but not reaped by its parent pick up

Meaning of additional characters:

- $<$: high-priority (not nice to other users)
- N : low-priority (nice to other users)
- L : has pages locked into memory (for real-time and custom IO)
- s : is a session leader
- l : is multi-threaded
- $+$: is in the foreground process group

Home Work

1. Find in total how many processes are running in your system?
2. Find the states of some processes.
3. Figure out the memory layout of any of your C program.

References:

- P. B. Galvin A. Silbeschatz and G. Gagne, ”Operating System Concepts”.
- Slides of Operating System Concepts