OS ASSIGNMENT-2 REPORT

Lab2- Work with processes

: Adarsh Anand : 2003101 :

: Prof - Dr Sharad Sinha : TA - Prachi Kashikar

QUESTIONS

Perform the following exercises:

1. Use the ps, ps lx, pstree and ps -aux command to display the process attributes.
2. 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. Compile a C program and observe the same effect (Use a long loop say while(1) to observe the

effect)

d. From the top display, answer the following: – How much memory is free in the system? – Which process is taking more CPU? – Which process has got maximum memory share?

e. Write a CPU bound C program and a I/O bound C program (e.g. using more printf statements within while(1) loop), compile and execute both of them. Observe the effect of their CPU share using the top display and comment.

3. Write a program in C that creates a child process, waits for the termination of the child and lists its PID, together with the state in which the process was terminated (in decimal and hexadecimal)

4. In a C program, print the address of the variable and enter into a long loop. Start three to four processes of the same program and observe the printed address values. Show how two processes which are members of the relationship parent child are concurrent from execution point of view, initially the child is copy of the parent, but every process has its own data.

ANSWERS

Perform the following exercises:

1. **Use the ps, ps lx, pstree and ps -aux command to display the process attributes.**

Ans) ps - snapshot of current process

```
PS(1)

NAME
       ps - report a snapshot of the current processes.
```

```
alpha@alpha-HP:~$ ps
    PID TTY          TIME CMD
 125732 pts/0    00:00:00 bash
 125749 pts/0    00:00:00 ps
alpha@alpha-HP:~$
```

ps lx

```
alpha@alpha-HP:~$ ps lx
F   UID     PID    PPID PRI  NI    VSZ   RSS WCHAN  STAT TTY        TIME COMMAND
4  1000    1304       1  20   0  19908 10644 ep_pol Ss   ?          0:00 /lib/systemd/systemd --user
5  1000    1305    1304  20   0 169600  3552 -      S    ?          0:00 (sd-pam)
0  1000    2858    1304   9 -11 2751368 18400 do_sys S<sl ?          0:00 /usr/bin/pulseaudio --daemonize=no --log-target=journal
0  1000    2860    1304  39   - 511556 25048 do_sys SNsl ?          0:00 /usr/libexec/tracker-miner-fs
1  1000    2863       1  20   0 240308  7808 -      SLl  ?          0:00 /usr/bin/gnome-keyring-daemon --daemonize --login
4  1000    2869    1267  20   0 164016  6564 do_sys Ssl+ tty2       0:00 /usr/lib/gdm3/gdm-x-session --run-script env GNOME_SHELL_SESSION_MODE=ubuntu /usr/bin/gnome-session --systemd --sessi
0  1000    2870    1304  20   0   9148  6236 ep_pol Ss   ?          0:01 /usr/bin/dbus-daemon --session --address=systemd: --nofork --nopidfile --systemd-activation --syslog-only
4  1000    2873    2869  20   0 26054784 136552 ep_pol Sl+ tty2     0:47 /usr/lib/xorg/Xorg vt2 -displayfd 3 -auth /run/user/1000/gdm/Xauthority -background none -noreset -keeptty -verbose 3
0  1000    2883    1304  20   0 239704  7844 do_sys Ssl  ?          0:00 /usr/libexec/gvfsd
0  1000    2892    1304  20   0 378344  6172 futex_ Sl   ?          0:00 /usr/libexec/gvfsd-fuse /run/user/1000/gvfs -f -o big_writes
0  1000    2911    1304  20   0 314056  9484 do_sys Ssl  ?          0:00 /usr/libexec/gvfs-udisks2-volume-monitor
0  1000    2919    1304  20   0 316724  8756 do_sys Ssl  ?          0:00 /usr/libexec/gvfs-afc-volume-monitor
0  1000    2924    1304  20   0 236496  6760 do_sys Ssl  ?          0:00 /usr/libexec/gvfs-goa-volume-monitor
0  1000    2928    1304  20   0 962424 81556 do_sys SLl  ?          0:00 /usr/libexec/goa-daemon
0  1000    2936    1304  20   0 314624  8984 do_sys Sl   ?          0:00 /usr/libexec/goa-identity-service
0  1000    2942    1304  20   0 238120  6444 do_sys Ssl  ?          0:00 /usr/libexec/gvfs-gphoto2-volume-monitor
0  1000    2947    1304  20   0 235700  5984 do_sys Ssl  ?          0:00 /usr/libexec/gvfs-mtp-volume-monitor
0  1000    3063    1304  20   0 239896  7744 -      Sl   ?          0:00 /usr/bin/gnome-keyring-daemon --start --foreground --components=secrets
0  1000    3158    2869  20   0 188240 14048 do_sys Sl+  tty2       0:00 /usr/libexec/gnome-session-binary --systemd --systemd --session=ubuntu
1  1000    3241    3158  20   0   6040   452 -      Ss   ?          0:00 /usr/bin/ssh-agent /usr/bin/im-launch env GNOME_SHELL_SESSION_MODE=ubuntu /usr/bin/gnome-session --systemd --session=
0  1000    3259    1304  20   0 305412  6804 do_sys Ssl  ?          0:00 /usr/libexec/at-spi-bus-launcher
0  1000    3264    3259  20   0   7376  4324 ep_pol S    ?          0:00 /usr/bin/dbus-daemon --config-file=/usr/share/defaults/at-spi2/accessibility.conf --nofork --print-address 3
0  1000    3291    1304  20   0  90060  4376 do_sys Ssl  ?          0:00 /usr/libexec/gnome-session-ctl --monitor
0  1000    3297    1304  20   0 484196 15600 do_sys Ssl  ?          0:00 /usr/libexec/gnome-session-binary --systemd-service --session=ubuntu
0  1000    3311    1304  20   0 5872600 357644 do_sys Ssl ?         1:44 /usr/bin/gnome-shell
0  1000    3343    3311  20   0 310944  8336 do_sys Sl   ?          0:00 ibus-daemon --panel disable --xim
0  1000    3347    3343  20   0 236536  7032 do_sys Sl   ?          0:00 /usr/libexec/ibus-dconf
0  1000    3348    3343  20   0 272716 28660 do_sys Sl   ?          0:00 /usr/libexec/ibus-extension-gtk3
0  1000    3350    1304  20   0 193520 23844 do_sys Sl   ?          0:00 /usr/libexec/ibus-x11 --kill-daemon
0  1000    3352    1304  20   0 236372  6356 do_sys Sl   ?          0:00 /usr/libexec/ibus-portal
0  1000    3364    1304  20   0 162836  7420 do_sys Sl   ?          0:00 /usr/libexec/at-spi2-registryd --use-gnome-session
0  1000    3369    1304  20   0 235596  4620 do_sys Ssl  ?          0:00 /usr/libexec/xdg-permission-store
0  1000    3371    1304  20   0 1057284 22720 do_sys Sl  ?          0:00 /usr/libexec/gnome-shell-calendar-server
0  1000    3380    1304  20   0 943020 35092 do_sys Ssl  ?          0:00 /usr/libexec/evolution-source-registry
```

```
PSTREE(1)

NAME
       pstree - display a tree of processes
```

```
alpha@alpha-HP:~$ pstree
systemd─┬─ModemManager───2*[{ModemManager}]
        ├─NetworkManager───2*[{NetworkManager}]
        ├─accounts-daemon───2*[{accounts-daemon}]
        ├─acpid
        ├─anacron
        ├─atop
        ├─atopacctd
        ├─avahi-daemon───avahi-daemon
        ├─bluetoothd
        ├─colord───2*[{colord}]
        ├─cron
        ├─cups-browsed───2*[{cups-browsed}]
        ├─cupsd
        ├─dbus-daemon
        ├─fwupd───4*[{fwupd}]
        ├─gdm3─┬─gdm-session-wor─┬─gdm-x-session─┬─Xorg───4*[{Xorg}]
        │      │                 │               ├─gnome-session-b─┬─ssh-agent
        │      │                 │               │                 └─2*[{gnome-session-b}]
        │      │                 │               └─2*[{gdm-x-session}]
        │      │                 └─2*[{gdm-session-wor}]
        │      └─2*[{gdm3}]
        ├─gnome-keyring-d───3*[{gnome-keyring-d}]
        ├─irqbalance───{irqbalance}
        ├─2*[kerneloops]
        ├─networkd-dispat
        ├─nvidia-persiste
        ├─polkitd───2*[{polkitd}]
        ├─preload
        ├─rsyslogd───3*[{rsyslogd}]
        ├─rtkit-daemon───2*[{rtkit-daemon}]
        ├─snapd───19*[{snapd}]
        ├─switcheroo-cont───2*[{switcheroo-cont}]
        ├─systemd─┬─(sd-pam)
        │         ├─at-spi-bus-laun─┬─dbus-daemon
        │         │                 └─3*[{at-spi-bus-laun}]
        │         ├─at-spi2-registr───2*[{at-spi2-registr}]
        │         ├─2*[chrome_crashpad───2*[{chrome_crashpad}]]
        │         ├─chrome_crashpad───{chrome_crashpad}
        │         ├─cpptools-srv───15*[{cpptools-srv}]
        │         ├─dbus-daemon
        │         ├─dconf-service───2*[{dconf-service}]
        │         ├─evolution-addre───5*[{evolution-addre}]
        │         ├─evolution-calen───15*[{evolution-calen}]
        │         ├─evolution-sourc───3*[{evolution-sourc}]
        │         ├─gjs───10*[{gjs}]
```

ps -aux - Display all process

```
alpha@alpha-HP:~$ ps -aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.0  0.0 168496 11728 ?        Ss   15:02   0:01 /sbin/init splash
root           2  0.0  0.0      0     0 ?        S    15:02   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        I<   15:02   0:00 [rcu_gp]
root           4  0.0  0.0      0     0 ?        I<   15:02   0:00 [rcu_par_gp]
root           5  0.0  0.0      0     0 ?        I<   15:02   0:00 [netns]
root           7  0.0  0.0      0     0 ?        I<   15:02   0:00 [kworker/0:0H-events_highpri]
root          10  0.0  0.0      0     0 ?        I<   15:02   0:00 [mm_percpu_wq]
root          11  0.0  0.0      0     0 ?        S    15:02   0:00 [rcu_tasks_rude_]
root          12  0.0  0.0      0     0 ?        S    15:02   0:00 [rcu_tasks_trace]
root          13  0.0  0.0      0     0 ?        S    15:02   0:00 [ksoftirqd/0]
root          14  0.0  0.0      0     0 ?        I    15:02   0:03 [rcu_sched]
root          15  0.0  0.0      0     0 ?        S    15:02   0:00 [migration/0]
root          16  0.0  0.0      0     0 ?        S    15:02   0:00 [idle_inject/0]
root          17  0.0  0.0      0     0 ?        S    15:02   0:00 [cpuhp/0]
root          18  0.0  0.0      0     0 ?        S    15:02   0:00 [cpuhp/1]
root          19  0.0  0.0      0     0 ?        S    15:02   0:00 [idle_inject/1]
root          20  0.0  0.0      0     0 ?        S    15:02   0:00 [migration/1]
root          21  0.0  0.0      0     0 ?        S    15:02   0:00 [ksoftirqd/1]
root          23  0.0  0.0      0     0 ?        I<   15:02   0:00 [kworker/1:0H-events_highpri]
root          24  0.0  0.0      0     0 ?        S    15:02   0:00 [cpuhp/2]
root          25  0.0  0.0      0     0 ?        S    15:02   0:00 [idle_inject/2]
root          26  0.0  0.0      0     0 ?        S    15:02   0:00 [migration/2]
root          27  0.0  0.0      0     0 ?        S    15:02   0:00 [ksoftirqd/2]
root          29  0.0  0.0      0     0 ?        I<   15:02   0:00 [kworker/2:0H-events_highpri]
root          30  0.0  0.0      0     0 ?        S    15:02   0:00 [cpuhp/3]
root          31  0.0  0.0      0     0 ?        S    15:02   0:00 [idle_inject/3]
root          32  0.0  0.0      0     0 ?        S    15:02   0:00 [migration/3]
root          33  0.0  0.0      0     0 ?        S    15:02   0:00 [ksoftirqd/3]
root          35  0.0  0.0      0     0 ?        I<   15:02   0:00 [kworker/3:0H-events_highpri]
root          36  0.0  0.0      0     0 ?        S    15:02   0:00 [cpuhp/4]
root          37  0.0  0.0      0     0 ?        S    15:02   0:00 [idle_inject/4]
root          38  0.0  0.0      0     0 ?        S    15:02   0:00 [migration/4]
root          39  0.0  0.0      0     0 ?        S    15:02   0:00 [ksoftirqd/4]
root          41  0.0  0.0      0     0 ?        I<   15:02   0:00 [kworker/4:0H-events_highpri]
root          42  0.0  0.0      0     0 ?        S    15:02   0:00 [cpuhp/5]
root          43  0.0  0.0      0     0 ?        S    15:02   0:00 [idle_inject/5]
root          44  0.0  0.0      0     0 ?        S    15:02   0:00 [migration/5]
root          45  0.0  0.0      0     0 ?        S    15:02   0:00 [ksoftirqd/5]
root          47  0.0  0.0      0     0 ?        I<   15:02   0:00 [kworker/5:0H-events_highpri]
root          48  0.0  0.0      0     0 ?        S    15:02   0:00 [cpuhp/6]
root          49  0.0  0.0      0     0 ?        S    15:02   0:00 [idle_inject/6]
root          50  0.0  0.0      0     0 ?        S    15:02   0:00 [migration/6]
root          51  0.0  0.0      0     0 ?        S    15:02   0:00 [ksoftirqd/6]
root          53  0.0  0.0      0     0 ?        I<   15:02   0:00 [kworker/6:0H-events_highpri]
```

2. Learn the top command to display the resource utilization statistics of processes

a. Open a terminal and type the top command

Ans) Writing the top command shows the current running process.

```
TOP(1)

NAME
       top - display Linux processes

SYNOPSIS
       top -hv|-bcEHiOSs1 -d secs -n max -u|U user -p pid -o fld -w [cols]

       The traditional switches `-' and whitespace are optional.
```

```
top - 16:04:19 up  1:01,  1 user,  load average: 0.69, 0.46, 0.27
Tasks: 331 total,   2 running, 329 sleeping,   0 stopped,   0 zombie
%Cpu(s):  1.6 us,  1.3 sy,  0.0 ni, 96.9 id,  0.1 wa,  0.0 hi,  0.1 si,  0.0 st
MiB Mem :  15766.9 total,   4812.9 free,   3844.7 used,   7109.2 buff/cache
MiB Swap:   2048.0 total,   2048.0 free,      0.0 used.   9993.5 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   3311 alpha     20   0 5914828 393860 124816 R   8.6   2.4   2:00.07 gnome-shell
   2873 alpha     20   0   24.8g 136552  82992 S   5.6   0.8   0:56.46 Xorg
    229 root     -51   0       0      0      0 D   4.3   0.0   0:11.94 irq/148-SYNA32A
   4241 alpha     20   0 1124.9g 132880  95684 S   2.0   0.8   0:07.38 chrome
   1121 root      20   0  334428  14624  12652 S   1.3   0.1   0:04.15 touchegg
   3915 alpha     20   0   32.3g 110272  87512 S   0.7   0.7   0:22.66 chrome
  86604 root      20   0       0      0      0 I   0.7   0.0   0:01.12 kworker/u16:1-events_unbound
   3364 alpha     20   0  162836   7420   6636 S   0.3   0.0   0:00.49 at-spi2-registr
   3582 alpha     20   0  361444  43568  25004 S   0.3   0.3   0:04.65 indicator-cpufr
   3598 alpha     20   0   32.6g 366700 185428 S   0.3   2.3   2:05.36 chrome
   4281 alpha     20   0 1124.9g 196768  95460 S   0.3   1.2   0:14.74 chrome
  15781 alpha     20   0 1686596  33360  13520 S   0.3   0.2   0:00.82 cpptools
  17612 alpha     20   0 7993872 306772 162124 S   0.3   1.9   0:09.46 soffice.bin
  46260 alpha     20   0 1125.0g 443120 140100 S   0.3   2.7   1:32.55 chrome
  85836 root      20   0       0      0      0 I   0.3   0.0   0:00.29 kworker/1:2-mm_percpu_wq
  97532 root      20   0       0      0      0 I   0.3   0.0   0:01.36 kworker/4:2-events
 126623 root      20   0       0      0      0 I   0.3   0.0   0:00.42 kworker/u16:0-events_unbound
      1 root      20   0  168496  11728   8348 S   0.0   0.1   0:01.03 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
      5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 netns
      7 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/0:0H-events_highpri
     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   0:00.00 rcu_tasks_rude_
     12 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tasks_trace
     13 root      20   0       0      0      0 S   0.0   0.0   0:00.05 ksoftirqd/0
     14 root      20   0       0      0      0 I   0.0   0.0   0:03.66 rcu_sched
     15 root      rt   0       0      0      0 S   0.0   0.0   0:00.01 migration/0
     16 root     -51   0       0      0      0 S   0.0   0.0   0:00.00 idle_inject/0
     17 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
     18 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/1
     19 root     -51   0       0      0      0 S   0.0   0.0   0:00.00 idle_inject/1
     20 root      rt   0       0      0      0 S   0.0   0.0   0:00.09 migration/1
     21 root      20   0       0      0      0 S   0.0   0.0   0:00.03 ksoftirqd/1
     23 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/1:0H-events_highpri
     24 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/2
     25 root     -51   0       0      0      0 S   0.0   0.0   0:00.00 idle_inject/2
```

b. Start a browser and see the effect on the top display

Ans) Browsing Chrome has the highest CPU utilization among all processes.

```
top - 16:04:31 up  1:02,  1 user,  load average: 0.81, 0.50, 0.28
Tasks: 334 total,   1 running, 333 sleeping,   0 stopped,   0 zombie
%Cpu(s):  3.4 us,  1.6 sy,  0.0 ni, 95.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :  15766.9 total,   4629.2 free,   4016.2 used,   7121.5 buff/cache
MiB Swap:   2048.0 total,   2048.0 free,      0.0 used.   9810.4 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   3598 alpha     20   0   32.6g 370144 185452 S  15.9   2.3   2:07.82 chrome
   3311 alpha     20   0 5872760 393876 124816 S   6.3   2.4   2:01.38 gnome-shell
   2873 alpha     20   0   24.8g 136552  82992 S   3.0   0.8   0:57.17 Xorg
 132736 alpha     20   0 1124.9g 154240  97656 S   3.0   1.0   0:01.54 chrome
    229 root     -51   0       0      0      0 D   2.0   0.0   0:12.11 irq/148-SYNA32A
 132693 alpha     20   0 1128.9g 109660  82952 S   2.0   0.7   0:00.27 chrome
  46260 alpha     20   0 1125.0g 522400 149764 S   1.7   3.2   1:35.35 chrome
   1167 root      35  15   12036   9032   2424 S   1.0   0.1   0:10.47 preload
 126623 root      20   0       0      0      0 I   1.0   0.0   0:00.48 kworker/u16:0-events_unbound
   3909 alpha     20   0   32.9g 225424 134300 S   0.7   1.4   1:28.33 chrome
   3915 alpha     20   0   32.3g 110284  87516 S   0.7   0.7   0:22.74 chrome
 132691 alpha     20   0 1128.9g  87748  67476 S   0.7   0.5   0:00.07 chrome
     14 root      20   0       0      0      0 I   0.3   0.0   0:03.68 rcu_sched
    255 root      20   0       0      0      0 S   0.3   0.0   0:00.87 jbd2/nvme0n1p7-
    486 root     -51   0       0      0      0 S   0.3   0.0   0:00.91 irq/158-iwlwifi
   1117 root      20   0  273756   9360   8524 S   0.3   0.1   0:06.24 thermald
   1121 root      20   0  334428  14624  12652 S   0.3   0.1   0:04.20 touchegg
   3511 alpha     20   0  682488  27504  20516 S   0.3   0.2   0:00.29 gsd-power
  15500 alpha     20   0   40.4g 463004  63644 S   0.3   2.9   0:10.19 code
  15781 alpha     20   0 1686596  33360  13520 S   0.3   0.2   0:00.83 cpptools
  86604 root      20   0       0      0      0 I   0.3   0.0   0:01.14 kworker/u16:1-events_unbound
  97532 root      20   0       0      0      0 I   0.3   0.0   0:01.42 kworker/4:2-events
 125695 alpha     20   0  818632  54648  41800 S   0.3   0.3   0:03.18 gnome-terminal-
 128578 root      20   0       0      0      0 I   0.3   0.0   0:00.28 kworker/6:0-events
 130468 root      20   0       0      0      0 I   0.3   0.0   0:00.02 kworker/7:0-events
 132657 alpha     20   0   12124   4136   3188 R   0.3   0.0   0:00.07 top
      1 root      20   0  168496  11728   8348 S   0.0   0.1   0:01.03 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
      5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 netns
      7 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/0:0H-events_highpri
     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   0:00.00 rcu_tasks_rude_
     12 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tasks_trace
     13 root      20   0       0      0      0 S   0.0   0.0   0:00.05 ksoftirqd/0
     15 root      rt   0       0      0      0 S   0.0   0.0   0:00.01 migration/0
     16 root     -51   0       0      0      0 S   0.0   0.0   0:00.00 idle_inject/0
     17 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
```

c. Compile a C program and observe the same effect (Use a long loop say while(1) to observe the effect)

Ans) The C program takes a lot of CPU Usage .OS gives more priority and allocates maximum CPU and memory to those tasks.

d. From the top display, answer the following: – How much memory is free in the system? – Which process is taking more CPU? – Which process has got the maximum memory share?

Ans)

```
top - 16:36:05 up  1:33,  1 user,  load average: 0.75, 0.68, 0.65
Tasks: 326 total,   1 running, 325 sleeping,   0 stopped,   0 zombie
%Cpu(s):  1.2 us,  0.7 sy,  0.0 ni, 98.1 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :  15766.9 total,   5300.0 free,   3422.2 used,   7044.8 buff/cache
MiB Swap:   2048.0 total,   2048.0 free,      0.0 used.  10789.6 avail Mem
```

Free Memory - 5300MB

e. Write a CPU bound C program and a I/O bound C program (e.g. using more printf statements within the while(1) loop), compile and execute both of them. Observe the effect of their CPU share using the top display and comment.

Ans)

```c
// Function to print 100 prime numbers

#include <math.h>

#include <stdio.h>


int main() {

   int i, j, k, n, count = 0;

   for (i = 2; i <= 100; i++) {

       for (j = 2; j <= i; j++) {

           if (i % j == 0) {

               break;

           }

       }

       if (j == i) {

           printf("%d ", i);

           count++;

       }
```

```
    }

    printf("\n");

    printf("Total prime numbers: %d\n", count);

    return 0;

}
```

```
// Write simple input ouput program

#include <stdio.h>


int main() {

    // take file input

    FILE *fp;

    fp = fopen("input.txt", "w");

    fprintf(fp, "Hello World\n");

    fclose(fp);

    // take file output

    fp = fopen("output.txt", "r");

    char c;

    while ((c = fgetc(fp)) != EOF) {

        printf("%c", c);

    }

    fclose(fp);

    return 0;

}
```

Running first file - CPU Usage 204%

```
  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
 8463 alpha     20   0   54.5g 392184 117448 R 204.7   2.4   2:25.92 code
 9440 alpha     20   0   44.5g 159336  96164 R  50.8   1.0   0:28.96 code
 9663 alpha     20   0   36.4g  93968  56060 S  34.2   0.6   0:12.73 code
42161 alpha     20   0    2496    576    512 R  25.9   0.0   0:06.20 cpu
 8422 alpha     20   0   32.6g 122696  83232 S  14.6   0.8   0:31.49 code
```

Running second file - CPU Usage 85%

```
top - 01:00:10 up 13 min,  1 user,  load average: 2.70, 1.32, 0.75
Tasks: 328 total,   4 running, 324 sleeping,   0 stopped,   0 zombie
%Cpu(s): 14.1 us,  2.4 sy,  0.0 ni, 73.4 id, 10.0 wa,  0.0 hi,  0.1 si,  0.0 st
MiB Mem :  15766.9 total,   7486.8 free,   3477.6 used,   4802.4 buff/cache
MiB Swap:   2048.0 total,   2048.0 free,      0.0 used.  10665.3 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   8463 alpha     20   0   54.5g 382240 117500 S  85.0   2.4   2:52.69 code
   8422 alpha     20   0   32.6g 122444  83232 R  15.9   0.8   0:34.22 code
   3233 alpha     20   0 5406356 342940 118480 R  13.0   2.1   0:51.77 gnome-shell
   2789 alpha     20   0   24.6g 127424  77896 S   6.3   0.8   0:28.91 Xorg
```

3. Write a program in C that creates a child process, waits for the termination of the child and lists its PID, together with the state in which the process was terminated (in decimal and hexadecimal)

Ans)

```c
//  Write a program in C that creates a child process,

//  waits for the termination of the child

//  and lists its PID, together with the state in which the process

//  was terminated (in decimal and hexadecimal)

#include <stdlib.h>

#include <stdio.h>

#include <sys/types.h>

#include <sys/wait.h>

#include <stdlib.h>

#include <unistd.h>

int main()

{

 int status;    // status of the child process

 pid_t pid;     // process id of the child process
```

```
 pid = fork(); // creates a child process
 if (pid == 0)
 {
   // child process created
   printf("Child process created with PID: %d\n", getpid());
   exit(0);
 }
 else
 {
   printf("Parent process created with PID: %d\n", getpid());
   wait(&status);
// waits for the child process to terminate
   printf("Child process terminated with status %d - decimal & %x - hexadecimal\n",
status, status); // prints the status of the child process
 }
 return 0;
}
/*
 Important code:
 - pid: process id
 - status: status of the child process
 - fork(): creates a child process
 - wait(&status): waits for the child process to terminate
 - getpid(): returns the process id of the current process
 - getppid(): returns the process id of the parent process
 - exit(): terminates the current process
 */
```

4. In a C program, print the address of the variable and enter into a long loop.

a)Start three to four processes of the same program and observe the printed address values.

Ans) `// 4a). In a C program, print the address of the variable`

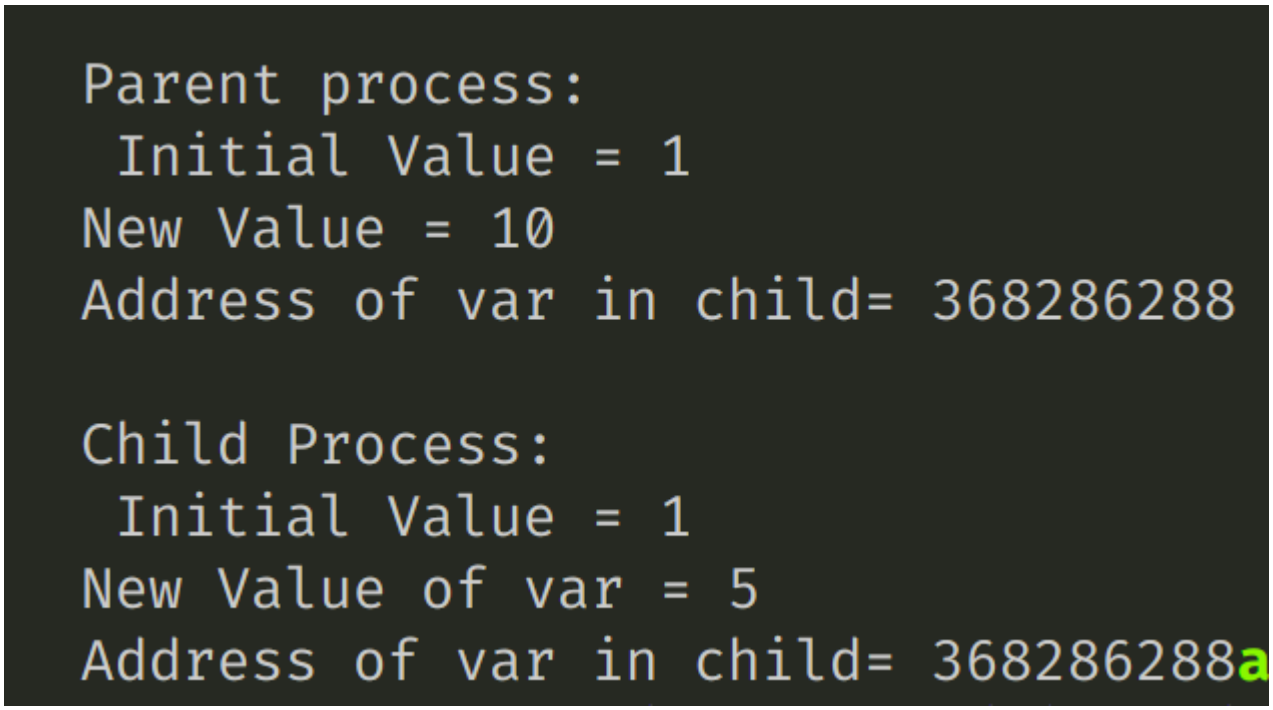
```
// and enter into a long loop.

// Start three to four processes of the same program

// and observe the printed address values.

// Show how two processes which are members of the relationship parent child are concurrent from execution point of view, initially the child is copy of the parent, but every process has its own data.

#include <stdio.h>

int main(){

 int a = 10;

 printf("%p\n", &a);

 while(1);

}
```

```
alpha@alpha-HP:~/Documents/GitHub/CS310-OS/Lab-2$ ./a.out
0x7ffe5b2c1494
alpha@alpha-HP:~/Documents/GitHub/CS310-OS/Lab-2$ ./a.out
0x7ffebf492be4
alpha@alpha-HP:~/Documents/GitHub/CS310-OS/Lab-2$ ./a.out
0x7ffc577eb9a4
alpha@alpha-HP:~/Documents/GitHub/CS310-OS/Lab-2$ ./a.out
0x7ffd8b014fe4
alpha@alpha-HP:~/Documents/GitHub/CS310-OS/Lab-2$
```

different addresses.

b) Show how two processes which are members of the relationship parent child are concurrent from execution point of view, initially the child is copy of the parent, but every process has its own data.

```c
#include <errno.h>
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <unistd.h>
int main(void) {
   int value = 1;
   pid_t PID = fork(); // create a child process
   if (PID >= 0) {
       if (PID == 0) { // child process
           printf("\n\nChild Process:\n Initial Value = %d", value);
           value = 5;
           printf("\nNew Value of value = %d", value);
           printf("\nAddress of value in child= %d", &value);
       } else { // parent process
           printf("\n\nParent process:\n Initial Value = %d", value);
           value = 10;
           printf("\nNew Value = %d", value);
           printf("\nAddress of value in child= %d", &value);
       }
   }
   return 0;
}
```



We can see each process has its own data. Hence proved that from execution point of view, initially the child is copy of the parent, but every process has its own data.

REFERENCES

- Dr Sharad Sir Slides
- C Programs given in .zip file
- Input-output system calls in C | Create, Open, Close, Read, Write - GeeksforGeeks
- Linux man pages
- Bash manual
- Wait System Call in C - GeeksforGeeks

END OF REPORT