



Mawlana Bhashani Science and Technology University

Lab-Report

Lab Report No: 03

Lab Report Name: How to view threads of a process on Linux and thread program .

Course code: ICT-3110

Course title: Operating System Lab

Date of Performance: 13-09-2020

Date of Submission:

Submitted by

Name: Md Masud Rana

ID: IT-18002

3rd year 1st semester

Session: 2017-18

Dept. of ICT.

Submitted To

Nazrul Islam

Assistant Professor

Dept. of ICT,

MBSTU.

Experiment No: 03

Experiment Name: How to view threads of a process on Linux and thread program .

Theory : A thread is a flow of execution through the process code, with its own program counter that keeps track of which instruction to execute next, system registers which hold its current working variables, and a stack which contains the execution history.

A thread shares with its peer threads few information like code segment, data segment and open files. When one thread alters a code segment memory item, all other threads see that.

A thread is also called a lightweight process. Threads provide a way to improve application performance through parallelism. Threads represent a software approach to improving performance of operating system by reducing the overhead thread is equivalent to a classical process.

Types of Thread: Threads are implemented in following two ways –

User Level Threads – User managed threads.

Kernel Level Threads – Operating System managed threads acting on kernel, an operating system core.

Code-

```
new.c
~/program

#include<stdio.h>
#include<string.h>
#include<pthread.h>
#include<stdlib.h>
#include<unistd.h>

pthread_t tid[2];

void* doSomething(void *arg)
{
    unsigned long i = 0;
    pthread_t id = pthread_self();

    if(pthread_equal(id,tid[0]))
    {
        printf("\n First thread processing\n");
    }
    else
    {
        printf("\n Second thread processing\n");
    }

    for(i=0; i<(0xFFFFFFFF);i++);

    return NULL;
}

int main(void)
{
    int i = 0;
    int err;

    while(i < 2)
    {
        err = pthread_create(&(tid[i]), NULL, &doSomething, NULL);
        if (err != 0)
            printf("\ncan't create thread :[%s]", strerror(err));
        else
            printf("\n Thread created successfully\n");

        i++;
    }

    sleep(5);
    return 0;
}
```

Output-

```
masud@masud-VirtualBox:~/program$ gcc new.c -lpthread
masud@masud-VirtualBox:~/program$ ./a.out

Thread created successfully

Thread created successfully

Second thread processing

First thread processing
masud@masud-VirtualBox:~/program$
```

Thread in command line:

Here are several ways to show threads for a process on Linux.

1: **PS**- In ps command, "-T" option enables thread views. The following command list all threads created by a process with <pid>.

```
masud@masud-VirtualBox:~/program$ ps -T -p 1197
  PID  SPID  TTY      TIME  CMD
 1197  1197  tty2      01:06:55  gnome-shell
 1197  1199  tty2      00:00:00  gmain
 1197  1200  tty2      00:00:14  gdbus
 1197  1201  tty2      00:00:00  dconf worker
 1197  1210  tty2      00:00:00  threaded-ml
 1197  1211  tty2      00:00:00  JS Helper
 1197  1212  tty2      00:00:00  JS Helper
 1197  1213  tty2      00:00:00  JS Helper
 1197  1214  tty2      00:00:00  JS Helper
 1197  1215  tty2      00:00:00  JS Helper
```

2: **Top**: The top command can show a real-time view of individual threads. To enable thread views in the top output, invoke top with "-H" option. This will list all Linux threads.

```
masud@masud-VirtualBox:~/program$ top

top - 23:29:02 up 1 day, 3:26, 1 user, load average: 0.28, 0.18, 0.11
Tasks: 219 total, 1 running, 185 sleeping, 0 stopped, 0 zombie
%Cpu(s): 14.1 us, 1.3 sy, 0.0 ni, 84.5 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 4039732 total, 120344 free, 2732888 used, 1186500 buff/cache
KiB Swap: 2097148 total, 2097148 free, 0 used. 923688 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM     TIME+ COMMAND
14337 masud    20   0 3352404 519384 142408 S   7.9  12.9   2:37.60 Web Content
 1197 masud    20   0 3218588 603940  37548 S   5.0  15.0   67:20.01 gnome-shell
 1054 masud    20   0 736948 287368  77716 S   1.0   7.1   9:18.35 Xorg
14782 masud    20   0  44560   4404   3648 R   1.0   0.1   0:00.13 top
14027 masud    20   0 795932  37496  26812 S   0.7   0.9   0:07.05 gnome-terminal-
   848 gdm      20   0 2927268 151648  34096 S   0.3   3.8   1:00.52 gnome-shell
 1442 masud    20   0 221668   6956   5760 S   0.3   0.2   0:14.20 ibus-engine-sim
14204 masud    20   0 3454760 364708 185736 S   0.3   9.0   0:42.78 MainThread
 1 root      20   0  225740   8104   6240 S   0.0   0.2   0:27.07 systemd
```

To restrict the top output to a particular process and check all threads running inside the process: then we use `$ top -H -p <pid>`.

3: **Htop**: A more user-friendly way to view threads per process is via htop, an ncurses-based interactive process viewer. This program allows you to monitor individual threads in tree views.


```
CPU[||||||||||||||||||||||||||||| 58.4%] Tasks: 145, 468 thr; 1 running
Mem[||||||||||||||||||||||||||| 2.70G/3.85G] Load average: 0.07 0.12 0.09
Swp[|||| 0K/2.00G] Uptime: 1 day, 03:29:14
```

PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
14337	masud	20	0	3274M	456M	139M	S	43.6	11.6	2:46.61	/usr/lib/firefox/firefox -contenttp
1197	masud	20	0	3143M	590M	37548	S	10.7	15.0	1h07:26	/usr/bin/gnome-shell
14204	masud	20	0	3373M	357M	181M	S	2.7	9.1	0:44.00	/usr/lib/firefox/firefox -new-wind
1054	masud	20	0	719M	280M	77716	S	2.0	7.1	9:19.53	/usr/lib/xorg/Xorg vt2 -displayfd
14788	masud	20	0	34116	5056	3908	R	0.7	0.1	0:00.10	htop
14027	masud	20	0	777M	37496	26812	S	0.7	0.9	0:07.43	/usr/lib/gnome-terminal/gnome-term
1	root	20	0	220M	9184	6348	S	0.7	0.2	0:27.89	/sbin/init splash
1060	masud	20	0	719M	280M	77716	S	0.7	7.1	0:23.67	/usr/lib/xorg/Xorg vt2 -displayfd
14306	masud	20	0	2526M	104M	85244	S	0.7	2.6	0:00.28	/usr/lib/firefox/firefox -contenttp
14352	masud	20	0	3274M	456M	139M	S	0.0	11.6	0:07.51	/usr/lib/firefox/firefox -contenttp
14609	masud	20	0	3274M	456M	139M	S	0.0	11.6	0:06.35	/usr/lib/firefox/firefox -contenttp
14221	masud	20	0	3373M	357M	181M	S	0.0	9.1	0:00.28	/usr/lib/firefox/firefox -new-wind
14230	masud	20	0	3373M	357M	181M	S	0.0	9.1	0:00.17	/usr/lib/firefox/firefox -new-wind
14272	masud	20	0	3373M	357M	181M	S	0.0	9.1	0:06.12	/usr/lib/firefox/firefox -new-wind
14299	masud	20	0	3373M	357M	181M	S	0.0	9.1	0:00.77	/usr/lib/firefox/firefox -new-wind
209	root	19	-1	109M	17200	15676	S	0.0	0.4	0:03.44	/lib/systemd/systemd-journald
229	root	20	0	47052	5500	3164	S	0.0	0.1	0:01.87	/lib/systemd/systemd-udev
405	avahi	20	0	47208	3244	2904	S	0.0	0.1	0:00.14	avahi-daemon: running [masud-Virtu
406	avahi	20	0	47076	340	0	S	0.0	0.0	0:00.00	avahi-daemon: chroot helper
441	systemd-r	20	0	71680	6884	5144	S	0.0	0.2	0:06.24	/lib/systemd/systemd-resolved
447	systemd-t	20	0	142M	3104	2548	S	0.0	0.1	0:00.19	/lib/systemd/systemd-timesyncd
443	systemd-t	20	0	142M	3104	2548	S	0.0	0.1	0:01.00	/lib/systemd/systemd-timesyncd
527	root	20	0	4552	804	740	S	0.0	0.0	0:01.62	/usr/sbin/acpid
576	syslog	20	0	256M	3728	2960	S	0.0	0.1	0:00.24	/usr/sbin/rsyslogd -n
577	syslog	20	0	256M	3728	2960	S	0.0	0.1	0:00.04	/usr/sbin/rsyslogd -n
578	syslog	20	0	256M	3728	2960	S	0.0	0.1	0:00.28	/usr/sbin/rsyslogd -n
529	syslog	20	0	256M	3728	2960	S	0.0	0.1	0:00.62	/usr/sbin/rsyslogd -n
575	root	20	0	505M	8760	6244	S	0.0	0.2	0:00.02	/usr/lib/udisks2/udisksd
605	root	20	0	505M	8760	6244	S	0.0	0.2	0:00.01	/usr/lib/udisks2/udisksd
638	root	20	0	505M	8760	6244	S	0.0	0.2	0:00.00	/usr/lib/udisks2/udisksd
642	root	20	0	505M	8760	6244	S	0.0	0.2	0:00.00	/usr/lib/udisks2/udisksd
530	root	20	0	505M	8760	6244	S	0.0	0.2	0:00.45	/usr/lib/udisks2/udisksd
535	root	20	0	38760	3312	3020	S	0.0	0.1	0:00.20	/usr/sbin/cron -f
536	root	20	0	70700	5916	5124	S	0.0	0.1	0:01.86	/lib/systemd/systemd-logind
587	root	20	0	417M	5744	4384	S	0.0	0.1	0:00.00	/usr/sbin/ModemManager
606	root	20	0	417M	5744	4384	S	0.0	0.1	0:00.01	/usr/sbin/ModemManager
539	root	20	0	417M	5744	4384	S	0.0	0.1	0:00.16	/usr/sbin/ModemManager
555	messagebu	20	0	51988	6104	3676	S	0.0	0.2	0:17.34	/usr/bin/dbus-daemon --system --ad
651	root	20	0	569M	14156	10816	S	0.0	0.4	0:01.67	/usr/sbin/NetworkManager --no-daem
654	root	20	0	569M	14156	10816	S	0.0	0.4	0:00.47	/usr/sbin/NetworkManager --no-daem
604	root	20	0	569M	14156	10816	S	0.0	0.4	0:04.32	/usr/sbin/NetworkManager --no-daem

```
F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice F8Nice F9Kill F10Quit
```

Discussion: In this lab we learn how to view threads of process on linux. Here we use c language for creating threads. We also use command line to show the threads for a process on linux. For command line we use ps, top and Htop command. From this lab we also learn threads are implemented two ways such as user level threads and kernel level threads.