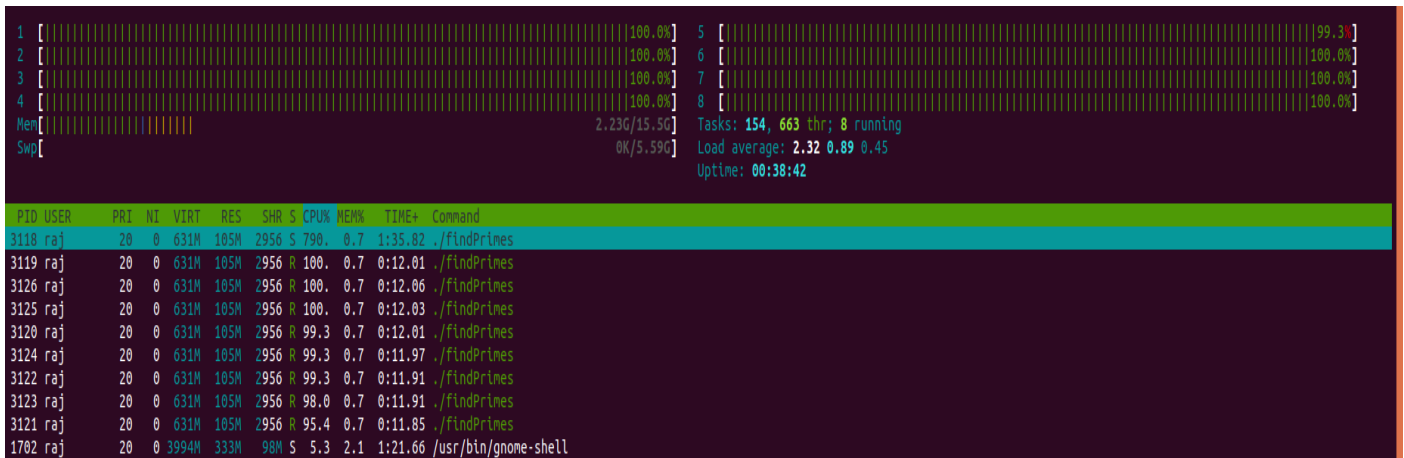


# Programming Assignment 1

## Description:

- This program finds the total number of prime numbers between 1 to  $10^8$  numbers using 8 concurrent threads. Once the process of finding prime numbers is complete, it also prints total execution time in seconds, sum of the prime numbers and top 10 maximum prime numbers found.
- The program discards 0, 1 and all of the even numbers except for 2. Hence, processing only odd values.
- It also makes use of atomic counter to safely cycle through all of the numbers from 1 to  $10^8$ . When saving the prime number, the program makes use of locking mechanism.
- Program does not involve any I/O access when threads are processing the numbers and only prints the required output (i.e. execution time, total number of primes, sum of primes, top 10 primes) once process completes.
- The prime number is found by calculating modulus till square root of a particular number.
- The experiments are run 6 times to provide an analysis of execution times. Screenshots of output are provided below as well as in file primes.txt. Execution time graph is also provided in the last section.
- Below is a screenshot of CPU usage which shows 8 threads running when findPrimes executable is run.



The screenshot shows a terminal window with a dark background. At the top, there are several progress bars for different processes, all showing 100.0% completion. Below these, there is a summary line: "Mem: 2.23G/15.5G", "Tasks: 154, 663 thr; 8 running", "Load average: 2.32 0.89 0.45", and "Uptime: 00:38:42". Below this, there is a table of running processes with columns: PID, USER, PRI, NI, VIRT, RES, SHR, S, CPU%, MEM%, TIME+, and Command. The table lists several processes running the command ". /findPrimes" with various resource usage statistics. At the bottom, there is a process for the user "raj" running the command "/usr/bin/gnome-shell".

PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
3118	raj	20	0	631M	105M	2956	S	790.0	0.7	1:35.82	./findPrimes
3119	raj	20	0	631M	105M	2956	R	100.0	0.7	0:12.01	./findPrimes
3126	raj	20	0	631M	105M	2956	R	100.0	0.7	0:12.06	./findPrimes
3125	raj	20	0	631M	105M	2956	R	100.0	0.7	0:12.03	./findPrimes
3120	raj	20	0	631M	105M	2956	R	99.3	0.7	0:12.01	./findPrimes
3124	raj	20	0	631M	105M	2956	R	99.3	0.7	0:11.97	./findPrimes
3122	raj	20	0	631M	105M	2956	R	99.3	0.7	0:11.91	./findPrimes
3123	raj	20	0	631M	105M	2956	R	98.0	0.7	0:11.91	./findPrimes
3121	raj	20	0	631M	105M	2956	R	95.4	0.7	0:11.85	./findPrimes
1702	raj	20	0	3994M	333M	98M	S	5.3	2.1	1:21.66	/usr/bin/gnome-shell

## Step in finding PrimeNumber:

1. Program sets a variable flag = "true" at the start of the process
2. Then each thread increments the atomic counter by 1 before calculating prime
3. the value is then calculated to find if it is a Prime number or not
4. if the value is not prime number flag value is changed to "false"
5. prime numbers are stored into a vector if flag = "true"

## How to run:

- program is written in c++ programming language with c++11 version and executed on Linux (Ubuntu) system.
- First unzip files. To run the code simply copy and paste following commands in terminal.

```
g++ -std=c++11 -pthread findPrimes.cpp -o findPrime
./findPrime
```

- First command will create an executable called “findPrimes” and second command will run this executable.

## Output Screenshots:

```
raj@raj-Lenovo:~$ ./findPrime-2
Execution Time : 500.372 seconds

number of primes found: 5761455

Sum of Prime numbers : 2556408908

Top 10 prime numbers:

99999787      99999821      99999827      99999839      99999847      99999931      99999941      99999959      99999971      99999989
raj@raj-Lenovo:~$ ./findPrime-2
Execution Time : 502.782 seconds

number of primes found: 5761455

Sum of Prime numbers : 2556408908

Top 10 prime numbers:

99999787      99999821      99999827      99999839      99999847      99999931      99999941      99999959      99999971      99999989
```

```
raj@raj-Lenovo:~$ ./findPrime-2
Execution Time : 487.845 seconds

number of primes found: 5761455

Sum of Prime numbers : 2556408908

Top 10 prime numbers:

99999787      99999821      99999827      99999839      99999847      99999931      99999941      99999959      99999971      99999989
raj@raj-Lenovo:~$ ./findPrime-2
Execution Time : 494.511 seconds

number of primes found: 5761455

Sum of Prime numbers : 2556408908

Top 10 prime numbers:

99999787      99999821      99999827      99999839      99999847      99999931      99999941      99999959      99999971      99999989
raj@raj-Lenovo:~$ ./findPrime-2
Execution Time : 498.934 seconds

number of primes found: 5761455

Sum of Prime numbers : 2556408908

Top 10 prime numbers:

99999787      99999821      99999827      99999839      99999847      99999931      99999941      99999959      99999971      99999989
raj@raj-Lenovo:~$ ./findPrime-2
Execution Time : 497.84 seconds

number of primes found: 5761455

Sum of Prime numbers : 2556408908

Top 10 prime numbers:

99999787      99999821      99999827      99999839      99999847      99999931      99999941      99999959      99999971      99999989
```

- It's clear from output that program finds 5761455 prime numbers which is the correct number of primes between 1 to  $10^8$ . The process takes approximately 495 seconds.

Execution time graph:

