Q3)

a)

Time taken for concurrent client handling using fork() system call

```
Factorial of 20 is 2432902008176640000
Time taken for 10 clients is 0.041328 seconds
pritish@LAPTOP-ECCOV209:/Assignment2$
```

Time Taken for concurrent client handling using pthreads

```
Factorial of 20 is 2432902008176640000
Time taken for 10 clients is 0.011909 secondspritish@LAPTOP-ECCOV209:/Assig
nment2$
```

Time Taken for concurrent client handling using select() system call

```
Factorial of 20 is 2432902008176640000
Time taken for 10 clients is 0.013157 secondspritish@LAPTOP-ECCOV209:/Assig
nment2$
```

Time Taken for concurrent client handling using poll() system call

```
Factorial of 20 is 2432902008176640000
Time taken for 20 clients is 0.012707 seconds
pritish@LAPTOP-ECCOV209:/Assignment2$
```

Time Taken for concurrent client handling using epoll() API

```
For client 9 Factorial of 20 is 2432902008176640000
Time taken for 20 clients is 0.082364 seconds
pritish@LAPTOP-ECCOV209:/Assignment2$
```

b) and c)

CPU Usage and Memory Usage for concurrent client handling using fork() system call

```
pritish@LAPTOP-ECCOV209:/Assignment2$ ./serverb
The pid of the server is 22102
```

```
  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
22102 pritish   20   0    2772    944    852 S   0.0   0.0   0:00.00 serverb
```

CPU Usage and Memory Usage for concurrent client handling using pthreads

```
pritish@LAPTOP-ECCOV209:/Assignment2$ ./serverc
The pid of the server is 23029
```

```
  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
23029 pritish   20   0  101220   2168   2012 S   0.0   0.0   0:00.03 serverc
```

CPU Usage and Memory Usage for concurrent client handling using select() system call

```
pritish@LAPTOP-ECCOV209:/Assignment2$ ./selectserver
The pid of the server is 23820
```

```
  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
23820 pritish   20   0    2772    960    864 S   0.0   0.0   0:00.02 selectserver
```

CPU Usage and Memory Usage for concurrent client handling using poll() system call

```
pritish@LAPTOP-ECCOV209:/Assignment2$ ./pollservertemp
The pid of the server is 24587
```

```
  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
24587 pritish   20   0    2772    988    896 S   0.0   0.0   0:00.04 pollservertemp
```

CPU Usage and Memory Usage for concurrent client handling using epoll() API

```
pritish@LAPTOP-ECCOV209:/Assignment2$ ./epollserver
The pid of the server is 26066
```

```
  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
26066 pritish   20   0    2772    940    848 S   0.0   0.0   0:00.01 epollserver
```

Q4)

We observe the fastest running time is observed with pthreads then the second fastest is with poll system call and the slowest time is with epoll API. Meanwhile for all the methods to implement concurrency in the server we observe that the CPU and memory usage for all of these is coming as zero which is understandable as these process take very less memory and cpu compared to available on our system as current computers have processors with clock speed of many GHz and Memory of 16 GB or so.

From the time we can conclude that the pthread and poll are the best methods for enabling the server to handle concurrency. Poll is a good option as we don't need to manually clear the fd and it is faster also.

But it can be seen all of them have very low running time which are in milliseconds so it doesn't matters much whatever we use here to perform the required task to enable server for concurrent client handling.