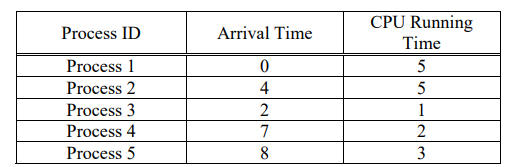
1. True or False:
   1. Threads that are part of the same process share the same stack pointer.
   2. The OS provides the illusion to each thread that it has its own address space.
   3. Threads that are part of the same process share the same general-purpose registers.
2. For the next questions, assume the following code is compiled and run on a modern Linux machine. Assume any irrelevant details have been omitted and that no routines, such as pthread\_create() or pthread\_join(), ever fail.

|  |
| --- |
| volatile int balance = 0;  void \*mythread(void \*arg) {  int result = 0;  result = result + 200;  balance = balance + 200;  printf(“Result is %d\n”, result);  printf(“Balance is %d\n”, balance);  return NULL; }  int main(int argc, char \*argv[]) {  pthread\_t p1, p2;  pthread\_create(&p1, NULL, mythread, “A”);  pthread\_create(&p2, NULL, mythread, “B”);  pthread\_join(p1, NULL);  pthread\_join(p2, NULL);  printf(“Final Balance is %d\n”, balance);  return 0;  } |

* 1. How many total threads are part of this process?
  2. When thread p1 prints “Result is %d\n”, what value of result will be printed?
  3. When “Final Balance is %d\n” is printed, what value of balance will be printed?

1. Here is a table of processes and their associated arrival and running times.



Show the scheduling order for these processes under 3 policies:

Even Roll number student: Shortest-Remaining-Time-First (SRTF) and Odd Roll number student: Round-Robin (RR) with timeslice quantum = 1. Find out the average waiting time and average response time.