1. True or False:
   1. Threads that are part of the same process share the same stack pointer.

False; each thread has its own stack and stack pointer so it can have own call stack and local.

* 1. The OS provides the illusion to each thread that it has its own address space.

False; each process, not thread, has its own address space.

* 1. Threads that are part of the same process share the same general-purpose registers. True

False; each thread must have own registers (saved and restored when that thread is not executing)

1. 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?

3: Main thread plus two created threads

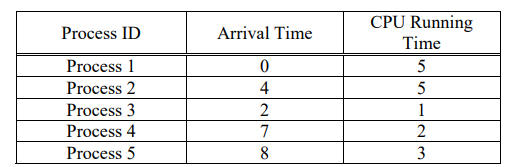
* 1. When thread p1 prints “Result is %d\n”, what value of result will be printed?

200: ‘result’ is a local variable allocated on the stack; each thread has its own private copy which only it increments, therefore there are no race conditions.

* 1. When “Final Balance is %d\n” is printed, what value of balance will be printed?

balance is allocated on the heap and shared between the two threads that are each accessing it without locks; there is a race condition.

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.

