

**CNG 334**

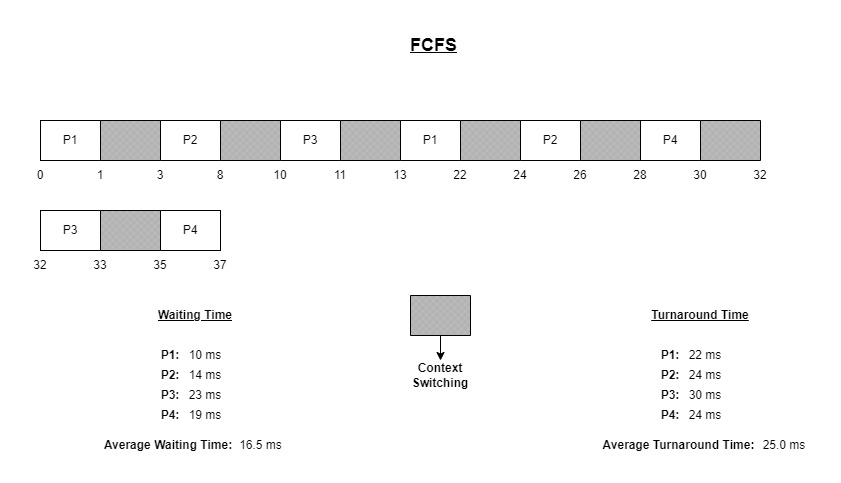
**Assignment 1 Report**

**Name:** Shayan Nadeem

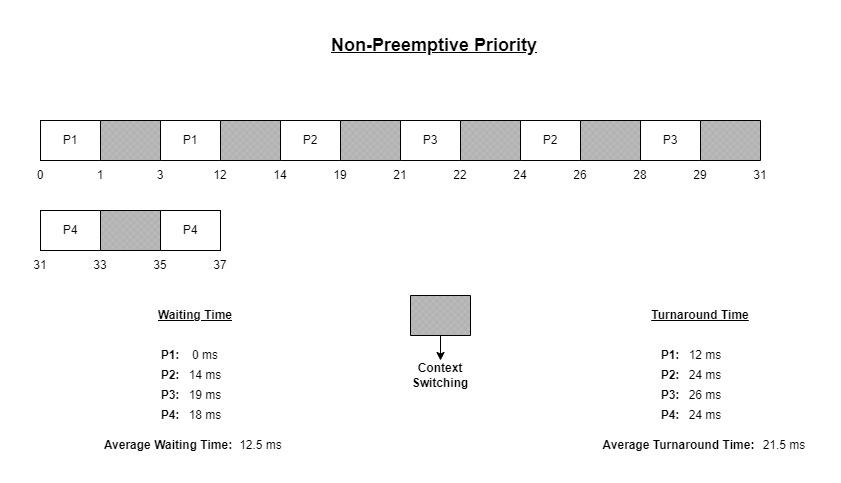
**Student ID:** 2542413

### **TASK 1:**

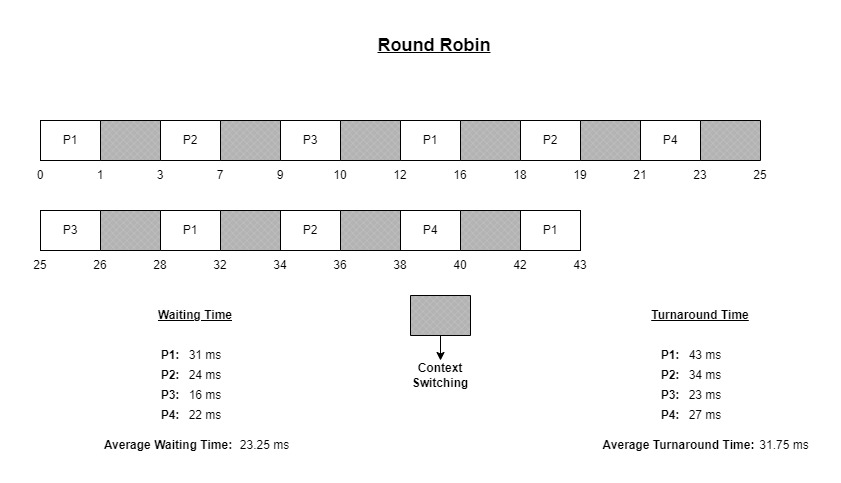
**1)**

****

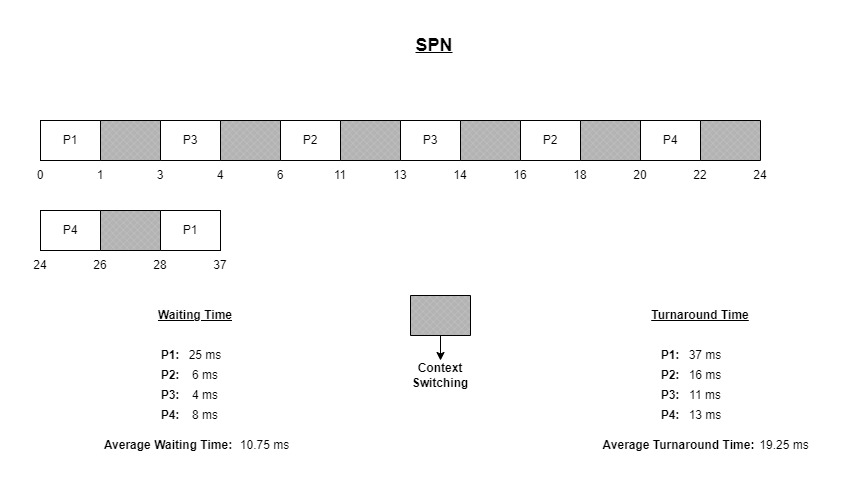
**2)**

****

**3)**

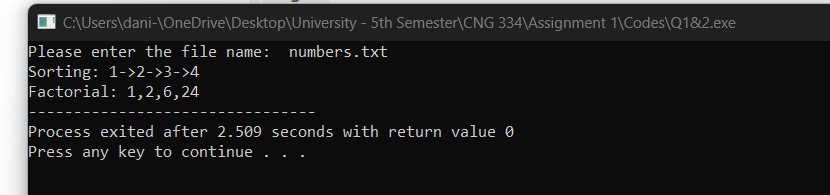
****

**4)**

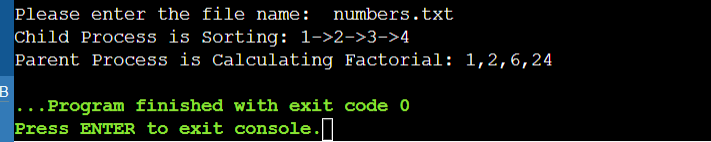
****

### **TASK 2:**

**Both Functions (readSort() & readCheck()) Output Screen:**

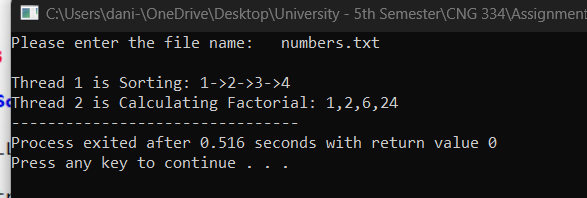
****

**Child And Parent Processes Output Screen:**

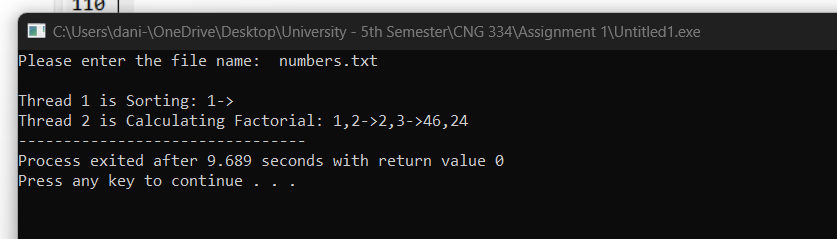
****

//I was not able to implement this in a normal DEVC++ compiler due to library issues so I had to use an online compiler for this part only//

**Threading Output Screen:**



**Threading (Including Waiting) Output Screen:**



### **TASK 3:**

**1:** Concurrency refers to the ability of multiple tasks to run in similar time periods, concurrently (one after another) but not simultaneously while parallelization involves the actual simultaneous execution of multiple tasks simultaneously using multiple CPU cores or threads. Multi Programming increases CPU utilization by allowing the CPU to switch between executing different tasks, which reduces idle time and increases overall efficiency. CPU is (in most cases) never allowed to stay IDLE. This will ultimately result in a better performance.

**2: Short Term Scheduling:** It is used to select processes to allocate CPU time to from an already **active (ready) queue of processes**. It maximizes the CPU utilization and minimizes the response time. Examples: SPN etc.

**Medium Term Scheduling:** Only goal is to retrieve processes from disk and putting them in memory or vice versa (in case of **I/O interrupts or blocking**). It is mainly used to resource optimization.

**Long Term Scheduling:** This is mostly done for Batch Processing Systems where **new processes** are selected to be loaded into the main memory. This type of scheduling ensures that the system is not overloaded with too many processes, and there is enough memory to accommodate them.