

The task for this assignment was to implement a simplified Dispatcher for an OS.

My dispatcher simulates dispatching using the first come first serve policy and changes states of processes admitted to the system respectfully. It also accumulates statistics about each process and prints them at the end of the simulation.

In the simulation, there is a mix of processes arriving in the system with each process falling into one of

three categories: CPU bound process, short process, or I/O bound process.

- All CPU bound processes require 1000 milliseconds of CPU time to complete and do not do any I/O.
- All short processes require 200 milliseconds of CPU time to complete and do not do any I/O.
- All I/O bound processes require 200 milliseconds of CPU time to complete and do I/O requests after 50, 100, and 150 of used CPU time.

Processes start to arrive in the system at time 0 and arrive every 100 milliseconds. Process ID is assigned to the processes in order of arrival, starting with ID = 1.

The mix of processes arriving in the system is defined by a single string.

For example, the string 'CCIIS' describes a mix of 5 processes:

- the process with ID=1 is CPU bound and arrives in the system at time 0.
- the process with ID=2 is CPU bound and arrives in the system at time 100.
- the process with ID=3 is I/O bound and arrives in the system at time 200.
- the process with ID=4 is I/O bound and arrives in the system at time 300.
- the process with ID=5 is a short process and arrives in the system at time 400.

My dispatcher keeps track of events and changes in the state of the processes, considering the following additional conditions:

- There is only one type of I/O request - request to the hard drive.
- The hard drive serves requests using FIFO (first in first out) policy.
- Every request to the hard drive takes 1000 milliseconds to complete.
- If there are no ready user processes, then process number 0 (system idle process) is running.
- If process 0 is running and a new process is created, or as the result of an event one of the blocked processes become ready (unblocked); this process will get CPU immediately.
- If some process is unblocked and another one is pre-empted at the same time, the unblocked process will get into the ready queue (or ready list) first.
- If some process is unblocked (pre-empted), and another one is admitted to the system at the same time, the unblocked (pre-empted) process will get into the ready queue (or ready list) first.
- If a process requests an exchange with the hard drive and CPU quantum expires at the same time, this process will get into the blocked (not ready) state.

When all lines of the input are processed, my dispatcher will print the following cumulative information about all processes admitted to the system during simulation:

<process id> <total time Running> <total time Ready> <total time Blocked>

For the system idle process only **<total time Running> <total time Ready>** is printed and it is assumed that process 0 was created at time 0.

For example, if the input is

ISC

The following output is expected:

0 2050 1400

1 200 250 3000

2 1000 0 0

3 200 100 0