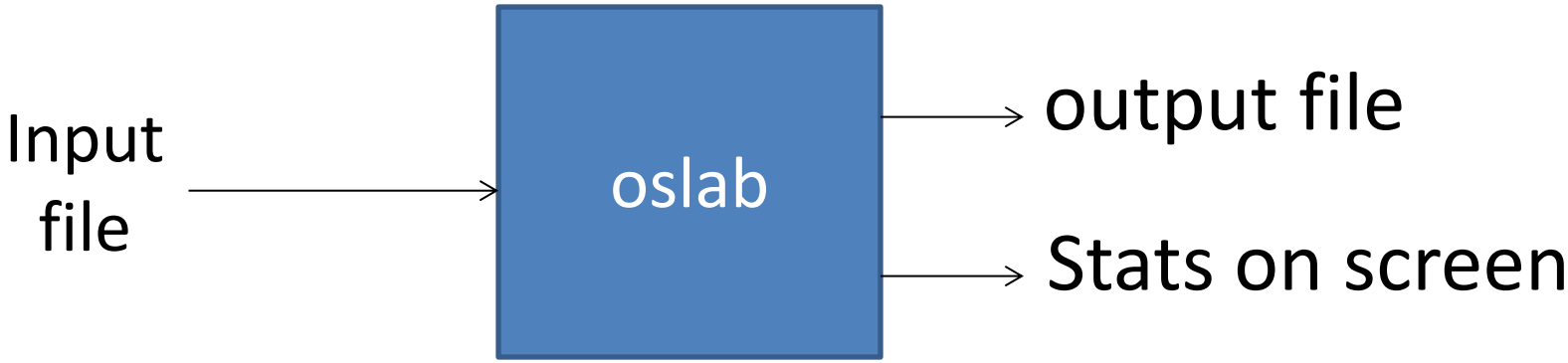
Programming Assignment Processes

What Will We Do?

In this assignment we will implement a simple OS multiprogramming environment on a single core.



./oslab filename

Your Source Code

- oslab.c
- compile with:

gcc –Wall –o oslab –std=c99 oslab.c

Input File

- The first line in the file is the total number of processes.
- This will be followed by x lines, each one presenting a process, where x = number of processes.
- Each process will be presented by 5 integers:

A B C D E

  - A: process ID
  - B: CPU time 1
  - C: I/O time
  - D: CPU time 2
  - Arrival time

Notes About Input

- CPU time 1 > 0
- IO time > 0
- CPU time 2 > 0
- Process ID are non-zero integers, less than 9999, and do not need to be consecutive.
  - No two processes will have the same ID.

You will Implement:
Simple Scheduler

- When there are several processes ready, the one assigned to the CPU (i.e. becomes “running”) is the one with lowest PID.

- If a process is running, and another process with lower PID becomes ready, the one with lower PID will take over.

- Given this, for this lab, time slice is not needed. Because a process is removed from the CPU only if:
  - It is done.
  - It becomes blocked for IO.
  - Another process with lower PID becomes available.

Output

- Output file name = input filename + num processes
  - Example: if input file name is “info” and the number of processes is 4, then the output file is “info4”.
- The format is the following:

```
time 0:
process : 76 : running
process : 89 : ready


time 1:
process : 76 : running
process : 89 : ready

…
```

Simple scheduling algorithm:
If more than one process are ready, the one with lowest PID is scheduled to run.

Example

Input file:

2
77 1 2 2 0
88 2 1 2 1

printed on the screen

Output statistics:

num processes = 2
CPU utilization = 1.000000
total time = 7
process 77: turnaround time = 4
process 88: turnaround time = 5

Output file:

time:0
process : 77 : running

time:1
process : 88 : running
process : 77 : blocked

time:2
process : 88 : running
process : 77 : blocked

time:3
process : 77 : running
process : 88 : blocked

time:4
process : 77 : running
process : 88 : ready

time:5
process : 88 : running

time:6
process : 88 : running

What To Submit

Your source code: single file with the name

oslab.c

One last thing

- To help you start, we are proving you with a C file (oslab.c) that:
  - Reads arguments from command line
  - Checks that the arguments are correct
  - Forms the name of the output file
  - Some helper code

- You can use this file, part of it, or none at all. It is up to you as long as your submitted program works correctly.

Testing your code

- We are providing you with an executable file (i.e. no compilation needed): oslabref
- After downloading it, type the following command before running it. You need to do this only once.
  - chmod 777 ./oslabref
- To use it: ./oslabref inputfile

To test your code

- Test your code with one process only.
- Then test with two processes that do not overlap.
- Then test with two processes that overlap.
- Then try with more sophisticated scenarios.
- Do not implement corner cases until you finish the regular one.
- We will not test your code with wrong inputs.

All the Best!