



Cairo University
Faculty of Engineering
Computer Engineering Department



OS Memory

Document Structure

- Requirement
- Guidelines
- Deliverables

Requirement

You are required to edit the *Scheduler* (in code file *scheduler.c*) from the “OS Scheduler” assignment (we may call it the *Core* from now on) to include memory allocation capabilities **using the buddy memory allocation system**. It should allocate memory space for processes as they enter the system and free it as they leave so that it can be re-used by later processes.

You may make the following assumptions,

- The total memory size is 1024 bytes.
- Each process size is less than or equal 256 bytes.
- Assume single uni-core CPU.
- The memory allocated for a process as it enters the system is constant over the time it spends in the system.

Input

You will need to (slightly) modify the the *Process Generator* (in code file *process_generator.c*) to accept an extra process information; *memsize*.

<i>processes.txt</i> example				
#id	arrival	runtime	priority	memsize
1	1	6	5	200
2	3	3	3	170

- Comments are added as lines beginning with `#` and should be ignored.
- Each non-comment line represents a process.
- Fields are separated with *one tab character* `'\t'`.
- You can assume that processes are sorted by their arrival time. *Take care that 2 or more processes may arrive at the same time.*

Output

A new output file *memory.log* should be generated for the memory information.

<i>memory.log</i> example	
#At time x allocated y bytes for process z from i to j	
At time 1 allocated 200 bytes for process 1 from 0 to 255	
At time 3 allocated 200 bytes for process 2 from 256 to 383	
At time 6 freed 200 bytes from process 2 from 256 to 383	
At time 10 freed 200 bytes from process 2 from 256 to 383	

- Comments are added as lines beginning with `#` and should be ignored.
- Make sure both *allocated* and *freed* memory information are logged.
- You need to stick to the given format because files are compared automatically.

Guidelines

- Read the document carefully at least once.
- Your program must not crash.
- You can use any IDE (Eclipse, Code::Blocks, NetBeans, KDevelop, CodeLite, etc.) you want of course, though it would be a good experience to use make files and standalone compilers and debuggers if you have time for that
- Spend a good time in design and it will make your life much easier in implementation.
- The code should be clearly commented and the variables names should be indicative.

Deliverables

You should deliver the modified code from the “OS Scheduler” assignment along with a modified version from the report that includes any necessary information about the memory allocation system described in this assignment.