

# Operating Systems

CMPT 424 • Fall 2018

## -iProject Three - 100 points

---

Goals	To build on the functionality of <i>iProject Two</i> (all of which is required) by adding the ability to execute <b>multiple</b> user programs at the same time.	
Functional Requirements	<input type="checkbox"/> Allow the user to load three programs into memory at once.	[5 points]
	<input type="checkbox"/> Add the following shell commands: <ul style="list-style-type: none"><li>• clearmem — clear all memory partitions</li><li>• runall — execute all programs at once</li><li>• ps — display the PIDs of all processes</li><li>• kill &lt;pid&gt; — kill a process</li><li>• quantum &lt;int&gt; — let the user set the Round Robin quantum (measured in cpu cycles)</li></ul>	[15 points]
Implementation Requirements	<input type="checkbox"/> Display the Ready queue and its contents (including process state) in real time.	[10 points]
	<input type="checkbox"/> [challenge] Track turnaround time and wait time for each process.	[+10 points]
	<input type="checkbox"/> Store multiple programs in memory, each in their own partition / segment, allocated by the client OS (which obviously needs to keep track of available and used partitions/segments).	[2 points]
	<input type="checkbox"/> Add base and limit registers to your core memory access code in the host OS and to your PCB objects in the client OS.	[2 points]
	<input type="checkbox"/> Enforce memory partition boundaries at all times.	[8 points]
	<input type="checkbox"/> Create a Resident list for the loaded processes.	[2 points]
	<input type="checkbox"/> Create a Ready queue for the running processes.	[2 points]
	<input type="checkbox"/> Instantiate a PCB for each loaded program and put it in the Resident list.	[2 points]
	<input type="checkbox"/> Develop a CPU scheduler in the client OS using Round Robin scheduling with the user-specified quantum measured in cpu cycles (default = 6). <ul style="list-style-type: none"><li>• Make the client OS control the host CPU with the CPU scheduler.</li><li>• Log all scheduling events.</li></ul>	[40 points]
	<input type="checkbox"/> Implement context switches with software interrupts. Be sure to update the mode bit (if appropriate), the PCBs, and the Ready queue.	[10 points]
	<input type="checkbox"/> Detect and gracefully handle errors like invalid op codes, missing operands (if you can detect that), and most importantly, memory out of bounds or access violation attempts.	[2 points]
<input type="checkbox"/> Your code must <i>separate structure from presentation</i> , be professionally formatted, use and demonstrate best practices, and make me proud to be your teacher.	[-∞ if not]	
<input type="checkbox"/> You must commit to Git early and often. I want to see many small and descriptive commits, not one or two massive ones. In fact, I will not accept projects with too few commits.	[-∞ if not]	

