Operating Systems

CMPT 424

Lab 011

Goals

Memory protection and executing many programs in memory
This active learning exercise will you help you make progress on the practical aspects of developing your operating system.

Instructions

1. Look at your *i*Project 3 functional requirements as Issues in GitHub as part of the "*i*Project 3" milestone and make sure that everything is in there.
2. Increase your memory from 256 bytes to 768 bytes. Be sure that you can map a memory partition number (0,1,2) to the appropriate base address (0, 256, 512).
3. Add memory protection fields (base and limit memory addresses) to your PCB.
4. Add to your Process Control Block as necessary to keep track of where a given process is held in memory.
5. If your memory is implemented properly and your context switching is working properly, then executing many programs in memory at once should be working just fine. But you and I both know that this is not the case. So I attached some test programs (next page) to help you debug. You're welcome.
6. Add the rest of the features as specified in your Issues and *i*Project 3.
7. Test. (You know this by now. Keep doing it.)
8. Read all of chapter 5 if you have not already done so. Even if you have already read it, read it again. It's probably my favorite chapter in our book. It's really good. Also read chapters 8.3 and 14.1 and 14.3.3 in the 8th edition of our text again. Good stuff!

Resources

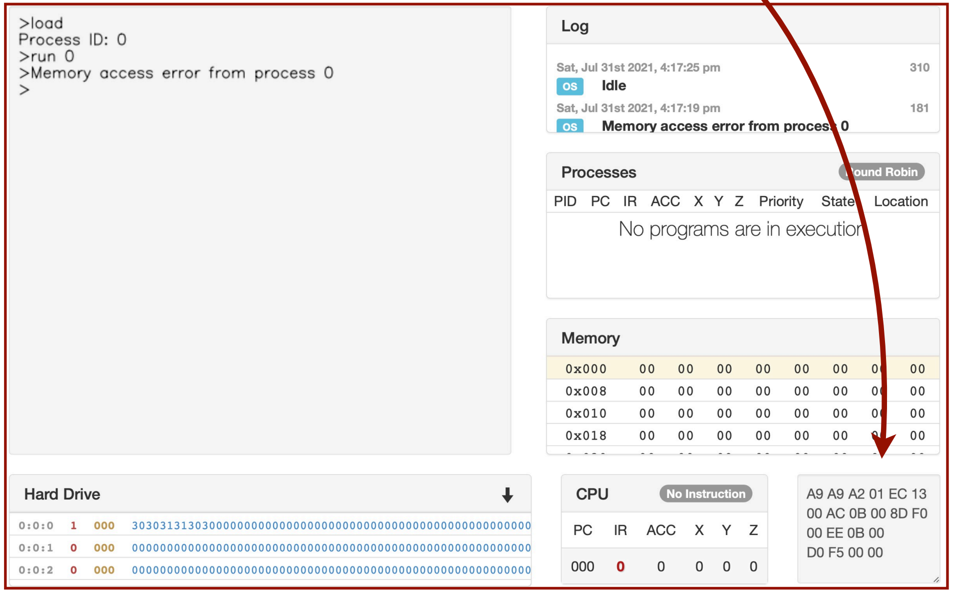
- http://lwn.net/Articles/250967/
- http://duartes.org/gustavo/blog/post/memory-translation-and-segmentation/
- Chapter 13 in http://pages.cs.wisc.edu/%7Eremzi/OSTEP/
- Chapter 15 in http://pages.cs.wisc.edu/%7Eremzi/OSTEP/

- Code to test memory limits:

  A9 A9 A2 01 EC 13 00 AC 0B 00 8D F0 00 EE 0B 00 D0 F5 00 00

Submitting

As usual.



```
>load
Process ID: 0
>run 0
>Memory access error from process 0
>
```

Log

```
Sat, Jul 31st 2021, 4:17:25 pm                310
OS  Idle
Sat, Jul 31st 2021, 4:17:19 pm                181
OS  Memory access error from process 0
```

Processes — Round Robin

PID	PC	IR	ACC	X	Y	Z	Priority	State	Location

No programs are in execution

Memory

```
0x000   00  00  00  00  00  00  00  00
0x008   00  00  00  00  00  00  00  00
0x010   00  00  00  00  00  00  00  00
0x018   00  00  00  00  00  00  00  00
```

Hard Drive

```
0:0:0  1  000  3030313130300000000000000000000000000000000000000000000000
0:0:1  0  000  0000000000000000000000000000000000000000000000000000000000
0:0:2  0  000  0000000000000000000000000000000000000000000000000000000000
```

CPU — No Instruction

PC	IR	ACC	X	Y	Z
000	0	0	0	0	0

```
A9 A9 A2 01 EC 13
00 AC 0B 00 8D F0
00 EE 0B 00
D0 F5 00 00
```

Operating Systems

CMPT 424

Test programs:

```
// a0a1a2adone
A9 00 8D 7B 00 A9 00 8D 7B 00 A9 00 8D 7C 00 A9 00 8D 7C 00 A9 01 8D 7A 00 A2 00 EC 7A 00 D0 39 A0 7D A2
02 FF AC 7B 00 A2 01 FF AD 7B 00 8D 7A 00 A9 01 6D 7A 00 8D 7B 00 A9 03 AE 7B 00 8D 7A 00 A9 00 EC 7A 00 D0
02 A9 01 8D 7A 00 A2 01 EC 7A 00 D0 05 A9 01 8D 7C 00 A9 00 AE 7C 00 8D 7A 00 A9 00 EC 7A 00 D0 02 A9 01 8D
7A 00 A2 00 EC 7A 00 D0 AC A0 7F A2 02 FF 00 00 00 00 61 00 61 64 6F 6E 65 00
```

```
// inner1 inner2 outer1 inner1 inner2 outer2 inner1 inner2 outer3
A9 00 8D EC 00 A9 00 8D EC 00 A9 00 8D ED 00 A9 00 8D ED 00 A9 00 8D EE 00 A9 00 8D EF 00 AD ED 00 8D FF 00
AE FF 00 A9 00 8D FF 00 EC FF 00 D0 BA AD EC 00 8D FF 00 A9 01 6D FF 00 8D EC 00 AD EC 00 8D FF 00 AE FF 00
A9 03 8D FF 00 EC FF 00 D0 05 A9 01 8D ED 00 A9 00 8D EE 00 A9 00 8D EF 00 AD EF 00 8D FF 00 AE FF 00 A9 00
8D FF 00 EC FF 00 D0 49 AD EE 00 8D FF 00 A9 01 6D FF 00 8D EE 00 AD EE 00 8D FF 00 AE FF 00 A9 02 8D FF 00
EC FF 00 D0 05 A9 01 8D EF 00 A9 F8 8D FF 00 A2 02 AC FF 00 FF AD EE 00 A2 01 8D FF 00 AC FF 00 FF A9 00 8D
FF 00 A2 01 EC FF 00 D0 A4 A9 F1 8D FF 00 A2 02 AC FF 00 FF AD EC 00 A2 01 8D FF 00 AC FF 00 FF A9 EE 8D FF
00 A2 02 AC FF 00 FF A9 00 8D FF 00 A2 01 EC FF 00 D0 33 00 00 00 20 20 00 20 6F 75 74 65 72 00 20 69 6E 6E 65
72 00 00
```

Look in the Hall of Fame operating systems for more testing ideas.