Todo

14 November 2014 21:07

1. Check Memory Manager (low-level) functions exist & init properly
   → Existing functions → Only do straight-up mapping
   + physical mem layout management → Need to create a stack of free pages (physical)
   + virtual mem layout management → Need to create a stack of free pages (virtual)
   → Need to separate out attribute setting
   → Need to create methods for:
   ↳ Finding free pages
   ↳ Add/remove from stacks
2. Basic process manager
   ↳ Load code
   ↳ Create process & single thread
   ↳ Start Scheduler
3. Basic Scheduler
   ↳ Switch register values, processes & threads
   ↳ Very simple fixed-time based system

Load Code / Create Process / Create Thread

1. Create process object
   ↳ Add to list of processes
   ↳ Process object needs to track memory info for now. (This will shift to Memory Layout later).
2. Allocate enough pages for code to be loaded
   ↳ 1 page for now.
   ↳ Code should contain a micro stack, stack init and main method = inf. loop with text output & busy wait
   ↳ Add page to processes pages list
3. Create thread object
   ↳ Set execution (instruction) pointer
   ↳ Add thread to Processes' list

Scheduler

1. On invoke:
   - Register timer interrupt
2. On timer interrupt:
   - Switch into kernel space
   - Store process / thread state
   - Get next thread
   - Load process / thread state
   - Switch into process space
   - Return to execution

Process Manager

1. Init:
   - Create process & thread for the kernel
   - Invoke scheduler
   - Fill in stacks

Full todo list:

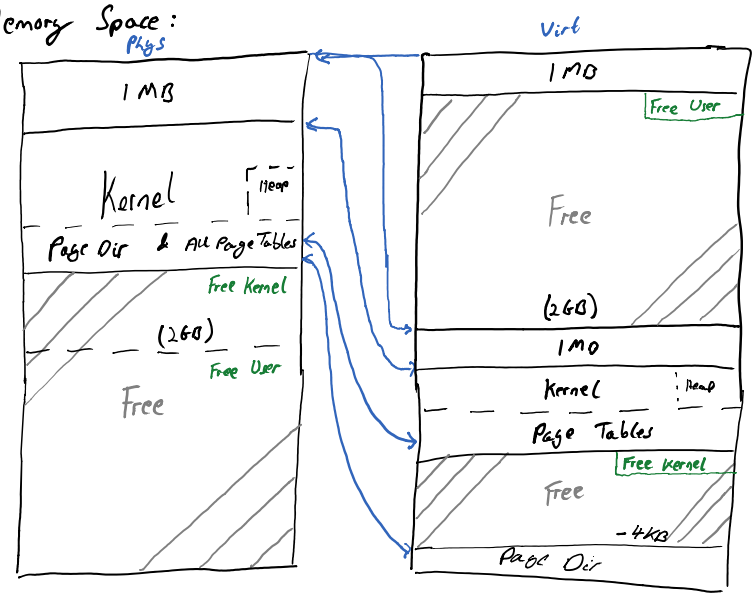
1. Set page attributes method
2. Create Stacks class
3. Create physical & virtual stacks
4. Add init code to fill in stacks
5. Check init mapping code
6. Create wrapper add/remove from stack methods
7. Create find free pages methods
8. Create map / unmap page methods
9. Create Process class
10. Create Thread class
11. Create Memory Layout class
12. Write Create/Init process/thread methods
13. Create Process Manager class
14. Add create process methods
15. Add init process memory methods
16. Add init thread objects
17. Create Scheduler class
18. Add scheduler methods
19. Add init scheduler calls
20. Create basic test program
21. Add code to invoke load of basic test program
22. Pray it all works!

Memory Manager

14 November 2014 20:07

Low-Level Manager – Manages physical memory layout & current virtual layout

Memory Space:



Init:
- Map reserved areas
- Self-map PD

Functions:
- Map Page
- Unmap Page
- Set Page Attributes

To: - Check Init functions
- Implement functions

Second-Level Manager – Handles multiple memory layouts (virtual only)

Functions:
- Create Layout
- Destroy Layout
- Load Layout
- Save Layout

Layout:
- PD & PTs

Process Manager

14 November 2014 21:00

Functions:

- Load code from file into memory
- Create a memory layout for the process
- Setup a thread for process
- Start process
- Handle process exit
- Destroy thread
- Destroy layout
- Remove code from memory

Clarification:

Process owns:

- Memory layout
- Threads

Threads share processes' memory layout.

Scheduler

14 November 2014 21:05

Switch = Save then Load

- Switch processes
  - ↳ Switch memory layouts
  - ↳ Switch register values
  - (↳ Switch execution level)
- Pulls process & thread info from process manager
  - ↳ Supplies memory layout info.

System Calls

14 November 2014 21:16

Examples:

- Sleep
- Process Exit
- Memory Request
- Pipe commands

Pipes Manager

14 November 2014 21:17