Preliminary Design Document , Operating Systems

Nilesh Kulkarni(110050007),
Rajlaxmi(110050087),
Sushant Hiray(110050009),
Deepali Adlakha(11d170020),
Sanchit Garg(110050035),
Vipul Harsh(110050034)

March 28, 2014

Abstract

The aim of the project is to implement the undersaid things. Allocation of swap space to a process, loading of code of a process in its swap space, allocation of page frames to processes, loading of initial page(s) in memory, raising of page faults during operation of a process, etc.

1 Tasks

1.1 Memory Management Unit

A Memory Management unit with the following functionalities.

1. Normal Page Table for translating Logical addresses to Physical addresses.
2. Raise Page fault interrupt and execute the VM routine for Page handling, in case of a missing Page table entry.
3. Page Table Entry will encapsulate following attributes: valid bit, dirty bit, reference bit(s)(required for LRU), *[1]sharing information(linked list of process).
4. Page Table will be stored in the OS's RAM
5. *Inverted Page Table

1.2 Virtual Memory Manager

1. Routine for Page fault Handling.
2. Handling Protection Issues
3. Process Creation: Copy on Write

[1]Things marked in asterisk would be implemented if time permits

4. Preferred Page replacement Policies : LRU using local replacement and global replacement.

5. *Implementing Working set for a process, along with promotion & demotion (initial allocation taken from config file)

1.3 Shared Memory Management

1. Allow Process to share pages among them.

2. Will especially come into play when process will Fork

3. Will be signalled by MMU if a particular shared page is written by a process, so that copy on write can be implemented.

1.4 Swap Space Manager

1. Keeping track of Swap Space

2. Support Page In & Page Out functionality

3. Process swaped in & swapped out when given calls by Medium Term Scheduler

4. Pages swapped in and swapped out when asked by the VM Manager

1.5 Medium Term Scheduler

1. Perform Swap in and Swap out of a process, blocked or dormant

2. Signal swap space manager to swap in & swap out a Process

2 Handling Events

2.1 Page Fault Event

1. Block the faulting process & estimate the time for page fault and create an interrupt accordingly to make the process "Ready".

2. MMU will signal the VM

3. VM will look up the replacement policy offered by the OS

4. Depending on whether the allocation strategy is global or local, it will iterate over all the page tables and get the page to be replaced .

5. The page should be swapped in, and the required page(which caused the page fault) to be swapped out.

6. Update page table entries accordingly(processes to be kept in mind)

2.2 Protection Violation

1. MMU will signal the VM
2. Process will be abort immediately on a violation of memory access

2.3 *Writing on Shared Memory

1. Create a copy of the page that was shared, and make the process that wants to write to point to it.
2. Update Page Table

3 Deadline

Expected to be completed by 15th April. Incase the Design is too ambitious please suggest.