Project - 4

Phase 3

Design Document

Vignesh Kulothungan ([vxk111430@utdallas.edu](mailto:vxk111430@utdallas.edu))

Date: 24th April, 2014

**Algorithm:**

* Idle process resides inside OS memory space.
* When a process is submitted, all pages are loaded to swap space memory. Only the first page is brought to main memory.
* When page fault occurs, a request with necessary information is sent to IO thread to copy page between memory and swap space.
* Page fault processing is done by the CPU like allocating new frame etc., in “init\_pagefault handler”.
* When age value becomes 0, copy back is delayed till that frame is selected to be replaced. If dirty bit is set, it is copied back to swap space later.
* “get\_data”, “get\_instruction” and “put\_data” use “compute\_address” function which converts offset to physical address (main memory address).
* The IO device thread uses a queue to receive requests. The thread waits on a semaphore. When a page fault is handled by main thread, it puts a request in the queue and does a semaphore post.
* The IO device thread then unlocks from semaphore and processes the request. After completing the request, it raises the page fault complete interrupt bit.
* Debug information is printed based on the observation mode.

**New Files:**

* **swapspace.c**

It implements the swap space manager.

* **IOdevice.c**

It implements the IO device thread manager.

**Functions Implemented:**

* **dump\_age\_vector()**: command = “a”;

It dumps the age values of all physical frames.

* **dump\_dirty\_bit()**: command = “d”;

It dumps the dirty bit values of all physical frames.

* **dump\_free\_list()**: command = “f”;

It dumps the free list of main memory.

* **dump\_IO()**: command = “i”;

It dumps the IO request queue.

* **dump\_swap\_free\_list()**:

It dumps the free list of main memory.

**Data Structures Used:**

Free List:

* Dynamic array. Array size dynamically allocated based on memSize and pageSize. (Array size = memSize/pageSize)
* If a frame is not allocated, the index corresponding to the unallocated frame stores -1. Else stores the process id to which it is allocated.

Page Table:

* Dynamically allocated page table of size “Number of Pages \* sizeof(int)”. If page is unallocated to a physical frame, it stores -1 in it. When a frame is allocated to a page, it stores the frame Number.

Main Memory and Swap Space:

* Dynamically allocated main memory of size “memSize”.
* Dynamically allocated swap space memory of size “swapSize”.