CS433 Operating Systems

Homework #3

Part III – Memory Management

1. Explain the difference between internal and external fragmentation. (4 pts)

Internal Fragmentation is when the process has allocated more memory than required, a few space will be left inside the block. It occurs when memory is divided into fixed size partitions.

External Fragmentation occurs when processes execute and then are swapped out of memory. The processes that replace them are smaller and this results in small blocks of wasted memory randomly throughout the entire memory block.

1. When Given six memory partitions of 300 KB, 600 KB, 350 KB, 200 KB, 750 KB, and 125 KB (in order), how would the first-fit, best-fit, and worst-fit algorithms place processes of size 115 KB, 500 KB, 358 KB, 200 KB, and 375 KB (in order)? Rank the algorithms in terms of how efficiently they use memory. (6 pts)

First fit:

115 KB -> 300KB

500 KB -> 600KB

358 KB -> 750KB

200 KB -> 350KB

375 KB -> 750KB

Best fit:

115 KB -> 125KB

500 KB -> 600KB

358 KB -> 750KB

200 KB -> 200KB

375 KB -> 750KB

Worst fit:

115 KB -> 750KB

500 KB -> 750KB

358 KB -> 600KB

200 KB -> 350KB

375 KB -> Internal fragmentation. waits for 750KB block or 600KB block memory to get free.

Efficiency

Best fit > First fit > Worst fit

1. On a system with paging, a process cannot access memory that it does not own. Why? How could the operating system allow access to other memory? Why should it or should it not? (6 pts)

An address on a paging system is an offset and a logical page number. The physical page is found by searching a table based on the logical page number to produce a physical page number. The operating system can limit a process to accessing only those physical pages allocated to the process because it controls the contents of the table. There is no way for a process to refer to a page it does not own because the page will not be in the page table. To allow such access, an operating system simply needs to allow entries for non-process memory to be added to the process’s page table. This is useful when two or more processes need to exchange data—they just read and write to the same physical addresses (which may be at varying logical addresses). This results in efficient interprocess communication.

1. Assuming a 1-KB page size, what are the page numbers and offsets for the following address references (provided as decimal numbers): (10 pts)
   1. 3085

Page Number: 3085/1024 = 3

Offset: 3085 mod 1024 = 13

* 1. 42095

Page Number: 42095/1024 = 41

Offset: 42095 mod 1024 = 111

* 1. 215201

Page Number: 215201/1024 = 210

Offset: 215201 mod 1024 = 161

* 1. 650000

Page Number: 650000/1024 = 634

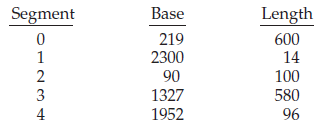
Offset: 650000 mod 1024 = 784

* 1. 2000001

Page Number: 2000001/1024 = 1953

Offset: 2000001 mod 1024 = 129

1. Consider the following segment table: Segment Base Length (5 pts)



What are the physical addresses for the following logical addresses?

1. 0,430

219 + 430 = 649

1. 1,10

2300 + 10 = 2310

1. 2,500

Illegal reference; traps to operating system

1. 3,400

1327 + 400 = 1727

1. 4,112

Illegal reference; traps to operating system

1. Assume that a program has just referenced an address in virtual memory. Describe a scenario in which each of the following can occur. (If no such scenario can occur, explain why.) (8 pts)

* TLB miss with no page fault
* TLB miss and page fault
* TLB hit and no page fault
* TLB hit and page fault

This can occur a scenario in which TLB miss with no page fault page has been brought into memory, but has been removed from the TLB.

1. What is the copy-on-write feature, and under what circumstances is its use beneficial? What hardware support is required to implement this feature? (5 pts)

Copy-on write is when two processes access the same set of program values. When this occurs, it is useful to map the corresponding pages into the virtual address spaces of the two programs in a write-protected manner. If a write takes place, a copy must be made to allow the two programs to individually access the different copies while at the same time the two programs cannot interfere with each other. Hardware support required is: on each memory access, the page table needs to be consulted to check if page is write protected. If write protected, a trap should occur and OS should resolve issue.

1. Consider the following page reference string: (6 pts)

7, 2, 3, 1, 2, 5, 3, 4, 6, 7, 7, 1, 0, 5, 4, 6, 2, 3, 0 ,1.

Assuming demand paging with three frames, how many page faults would occur for the following replacement algorithms?

* LRU replacement **18**
* FIFO replacement **17**
* Optimal replacement **13**

1. What is the cause of thrashing? How does the system detect thrashing? Once it detects thrashing, what can the system do to eliminate this problem? (5 pts)

Trashing is when underallocation of minimum number of pages required by a process occurs. This results in a continuous page fault. System can detect trashing by checking level of CPU utilization when compared to multiprogramming. It can be eliminated by reducing the use of multiprogramming.

1. Consider a demand-paging system with the following time-measured utilizations:

(10 pts)

CPU utilization 20%

Paging disk 97.7%

Other I/O devices 5%

For each of the following, indicate whether it will (or is likely to) improve

CPU utilization. Explain your answers.

1. Install a faster CPU.

This will not improve CPU utilization because a faster CPU reduces CPU utilization. The end result will be the CPU will spend more time waiting for processes to enter ready queue state.

1. Install a bigger paging disk.

This will not improve CPU utilization because the size of paging disk will not affect the amount of memory that is needed to reduce page faults.

1. Increase the degree of multiprogramming.

This will not improve CPU utilization because each process would have fewer frames available. Actually, the opposite will happen and the page fault rate would increase.

1. Decrease the degree of multiprogramming.

This will improve CPU utilization because other processes will have more frames and also reduce the rate of the page fault.

1. Install more main memory.

This will improve CPU utilization because other processes will have more memory and thus more pages can remain and will lessen the need to page to or from disks.