Jay Patel

CS 352

HW6

Professor Johnny Wang

8.13:

External fragmentation: Contiguous allocation with variable size partition does suffer from external fragmentation but Contiguous allocation with fixed size partitions does not. Segmentation does suffer from external fragmentation but pure paging does not suffer from external fragmentation since partitions and pages are fixed in size.

Internal fragmentation: Contiguous allocation with fixed size partitions and paging both may suffer from internal fragmentation when partitions and pages are not completely filled. Segmentation and variable sized partitions do not suffer from internal fragmentation since by definition, a partition or a segment is exactly as large as it needs to be.

8.20

Page number	Offset
3085/1024 = 3	3085 % 1024 = 13
42095/1024 = 41	42095 % 1024 = 111
215201/1024 = 210	215201 % 1024 = 161
650000/1024 = 634	650000 % 1024 = 784
2000001/1024 = 1953	2000001 % 1024 = 129

8.23

a. 256 pages = 2^8 pages, 12 + 8 = 20 bits

b. 64 frames = 2^6 frames, 12 +6 = 18 bits

9.21

a) 18 because whenever the frames are full and we want to access a new page we replace the lest recently used page with the required page.

b) 17 because whenever all the three frames are fill and we want to access new page we replace the page which came first into these frames with the required page.

c) 13 because whenever all the three frames are full and we want to access new page we replace the page which we are going to access last with the required page.

9.27

a) Install a faster CPU
No.
Reason: A faster CPU reduces the CPU utilization, and the CPU will spend more time waiting for a process to enter in the ready queue.
b) Install a bigger paging disk
No.
Reason: The size of the paging disk does not affect the amount of memory that is needed to reduce the page faults.
c) Increase the degree of multiprogramming
No.
Reason: Each process would have fewer frames available and the page fault rate would increase.
d) Decrease the degree of multiprogramming
Yes.
Reason: The processes will have more framers in order to bring their pages in them by suspending some of the processes. Hence that would result in reducing the page faults.
e) Install more main memory
Likely.
Reason: more pages can remain resident and do not require paging to or from the disks.
f) Install a faster hard disk or multiple controllers with multiple hard disks
No.

Reason: An improvement for as the disk bottleneck is removed by faster response and more throughput to the disk, the CPU will get more date more quickly and the paging disk is the problem not other disks.

g) Add pre paging to the page-fetch algorithms

Likely.

Reason: The CPU will get more data faster, so it will be more in use. This is only the case if the paging action is amenable. That means that more larger pages will mean info is retrieved in a timely manner (quicker).

h) Increase the page size.

No.

Reason: Increasing the page size will result in fewer page faults if data is being accessed sequentially. If data access is more or less random, more paging action could ensure because fewer pages can be kept in memory and more data is transferred per page fault. So, this change is as likely to decrease utilization as it is to increase it.

9.32

Under allocation of the minimum number of pages required by a process, forcing it to continuously page fault can be caused by thrashing. The system can detect thrashing by evaluating the level of CPU utilization as compared to the level of multiprogramming. But, it can be eliminated by reducing the level of multiprogramming.

9.34

The set of resident pages for a process might be underestimated, allowing a process to be scheduled even though all of its required pages are not resident which results in a large number of page faults when the Δ is set to a small value. Whereas, when it is set to a large value, then a process's resident set is overestimated and this may prevent many processes from being scheduled even though their required pages are resident. It is unlikely to generate page faults since its resident set has been overestimated when a process is scheduled.