Nichamon Han-idhikul 5610545684

8.1 Explain the difference between internal and external fragmentation

Internal fragmentation occurs when memory allocates to a process may be slightly larger than a request memory. The size difference is internal fragmentation

External fragmentation when processes are loaded and then removed from memory, hole or free memory space occurs into small pieces. The total space in memory is enough for a request but the available spaces are not contiguous. This problem can be fixed by using compaction

8.5 Compare the memory organization schemes of contiguous memory allocation, pure segmentation, and pure paging with respect to the following issues:

a) External fragmentation

Paging avoids external fragmentation by using compaction, but segmentation dose not
Contiguous allocation also avoids external fragmentation

b) Internal fragmentation

Internal fragmentation can appear in paging becouse frames are allocated as units. If the memory requirements of a process do not happen to coincide with page boundries, the last frame allocated may not be completely full.
Contiguous allocation and segmentation avoids this fragmentation by breaking physical memory into fixed-sized blocks.

c) Ability to share code across processes

Sharing generally requires that either paging or segmentation be used to provide small packets of information, but contiguous allocation does not.

Nichamon Han-idhikul 5610545684

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

	logical address (binary)	page number	offset
a) 3085	0000000000000000000011 0000001101	3	13
b) 42095	0000000000000000101001 0001101111	41	111
c) 215201	0000000000000011010010 0010100001	210	161
d) 650000	0000000000 001001111010 1100010000	634	784
e) 2000001	0000000000011110100001 0010000001	1953	129

(8.15) Consider a logical address space of 256 pages with a 4-KB page size, mapped onto a physical memory of 64 frames.

a) How many bits are required in the logical address?

Page size = $4 \times 1024 = 2^{12}$ → bits offset = 12

Address space $256 = 2^8$ pages → bits page number = 8

} Logical address requires 20 bits

b) How many bits are required in the physical address?

Frame size = Page size = 2^{12}

There are 64 frames. So, total = $64 \times 2^{12} = 2^6 \times 2^{12} = 2^{18}$

Therefore 18 bits are required in physical address

Nichamon Han-idhikul 5610545684

(9.1) Assume that a program has just referenced an address in VM. Describe a scenario in which each of following can occur (If no such scenario can occur, explain why)

- TLB miss with no page fault

  TLB doesn't contain this entry but it exists in page table (valid)

- TLB miss and page fault

  TLB doesn't contain this entry and it is either invalid in page table or it is not on a disk

- TLB hit and no page fault

  TLB contains this entry and it is valid in page table

- TLB hit and page fault

  This situation can never happen.

(9.8) Consider the following page reference string:

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 following replacement algorithms?

- LRU replacement

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

page fault = 18

- FIFO replacement

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

page fault = 17

- Optimal replacement

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

page fault = 14

Nichamon Han-idhikul 5610545684

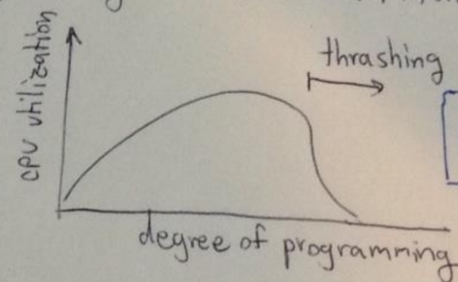
(9.14) Consider a demand-paging system with the following time-measured utilization:
CPU utilization 20%. Paging disk 97.7%. Other I/O devices 5%.
For each of the following, indicate whether it will (or is likely to) improve CPU utilization. Explain your answers.

a) Install a faster CPU No effect to CPU utilization

b) Install a bigger paging disk No effect to CPU utilization

c) Increase the degree of multiprogramming
CPU utilization increases slowly until a maximum is reached before thrashing. Then CPU utilization drops sharply.

d) Decrease the degree of multiprogramming
At maximum point, decrease degree of multiprogramming. CPU utilization increases and stop thrashing.



graph for c) and d)

e) Install more main memory
CPU utilization is improved because more pages can be stored in memory. Do not need to request page from disk.

f) Install a faster hard disk or multiple controllers with multiple hard disks.
Faster response in hard disk, CPU gets more data faster. Improve CPU utilization

g) Add prepaging to the page fetch algorithms.
Prepaging prevents high level of initial paging. CPU get more data faster. Improve CPU utilization.

h) Increase the page size
This result to fewer page fault if the access is sequential. If access is random, fewer pages keep in memory and data is transferred to page fault. So, it decreases CPU utilization

Nichamon Han-idhikul 5610545684

9.19 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?

Thrashing is caused by any process that does not have enough frames to support pages in active use. So, the page fault rate is high.

The system can detect thrashing with degree of multiprogramming and CPU utilization. The system reduces degree of multiprogramming to eliminate the problem.

11.3 What are the advantages of the variant of linked allocation that uses a FAT to chain together the blocks of a file

The benefit is that random-access time is improved, because the disk head can find the location of any block by reading the information in the FAT.

11.2 Contrast the performance of the three techniques for allocating disk blocks (Contiguous, linked, and indexed) for both sequential and random file access

Contiguous allocation

- sequential access : the file system remembers the disk address of the last block referenced and reads the next block when necessary. The difficulties are finding space for a new file and external fragmentation. It supports direct access.

Linked allocation

- sequential access : this allocation can be used effietively for this access. Need to start at the beginning of that file and follow pointers The space requires for the pointers. No external fragmentation
- random access : require FAT to improve access time.

Indexed allocation

- random access : this allocation brings all the pointers together into one location (index block). It suffers from wasted space. Dynamic access without external fragmentation

Nichamon Han-idhikul 5610545684

(11.8) Consider a file system uses inodes to represent files. Disk blocks are 8 KB in size, and a pointer to a disk block requires 4 bytes. This file system has 12 direct disk blocks, as well as single, double, and triple indirect disk blocks. What is the maximum size of a file that can be stored in this file system?

12 Disk blocks = 8 KB × 12 = 96 KB

$\frac{8KB}{4} = \frac{8 \times 1024 \text{ Bytes}}{4} = 2048$ Bytes

1 Single disk block = 2048 × 8KB
= 16384 KB

1 Double disk block = 2048 × 2048 × 8 KB
= 33554432 KB

1 Triple disk block = 2048 × 2048 × 2048 × 8KB
= 68719476736 KB

∴ Total size ≈ 64.03 TB

(12.2) Explain why SSDs often use an FCFS disk-scheduling algorithm.

The observerd behavior off SSDs indicates that the time required to service reads is uniform but that, because of the properties of flash memory, write service time is not uniform. Some SSD schedulers have exploited this property and merge only adjacent write request.

Nichamon Han-idhikul 5610545684

(12.3) Suppose that a disk drive has 5,000 cylinders, numbered 0 to 4,999. The drive is currently serving a request at cylinder 2,150 and the previous request was at cylinder 1,805. The queue of pending requests in FIFO order is:

2069 1212 2296 2800 544 1618 356 1523 4965 3681

Starting from the current head position, what is the total distance that the disk arm moves to satisfy all the pending requests for each of following disk-scheduling algorithms?

a) FCFS

2150 → 2069 → 1212 → 2296 → 2800 → 544 → 1618 → 356 → 1523 → 4965 → 3681

total distance = 13011

b) SSTF

2150 → 2069 → 2296 → 2800 → 3681 → 4965 → 1618 → 1523 → 1212 → 544 → 356

total distance = 7586

c) SCAN

2150 → 2296 → 2800 → 3681 → 4965 → 4999 → 2069 → 1618 → 1523 → 1212 → 544 → 356

total distance = 7492

d) LOOK

2150 → 2296 → 2800 → 3681 → 4965 → 2069 → 1618 → 1523 → 1212 → 544 → 356

total distance = 7397

e) C-SCAN

2150 → 2296 → 2800 → 3681 → 4965 → 4999 → 0 → 356 → 544 → 1212 → 1523 → 1618 → 2069

total distance = 9917

f) C-LOOK

2150 → 2296 → 2800 → 3681 → 4965 → 356 → 544 → 1212 → 1523 → 1618 → 2069

total distance = 9137

Nichamon Han-idhikul 5610545684

12.6 Describe some advantages and disadvantages of using SSDs as a caching tier and as a disk-drive replacement compared with using only magnetic disks.

SSD is much faster than megnetic disk drives because it has no moving parts and seek time or latency. It consumes less power. Some systems use SSD as a direct replacement for disk drives, while others use SSD as a new cache tier, moving data between megnetic disk, SSD and memory to optimize performance.

However, the cost of SSD is more expensive per megabyte than magnetic disk. Less capacity and shorter life spans.

13.1 When multiple interrupts from different device appear at about the same time, a priority scheme could be used to determine the order in which the interrupts would be serviced. Discuss what issues need to be considered in assigning priorities to different interrupt

First : Interrupts should be given higher priority than the trap that is given a low interrupt priority.

Second : Interrupts use to manage the flow of control within the kernel. The interrupts might be higher priority than interrupts of simply perform task.

Third : Devices that have real time constraints should be given higher priority than other devices. For example, some devices should be assigned higher priority, so data could be available for short time when the devices do not provide buffer.

Nichamon Han-idhikul 5610545684

13.4 User program needs a buffer for data transmition from a device. This buffer is specified by logical address or virtual address and it is in user space. Kernel will copy data from user buffer to its kernel buffer. Kernel can access user buffer by translate virtual address to physical address. The corresponding page need to obtain from swap space if userbuffer is not in physical memory. Then, the kernel can do I/O operation

13.5 When executing process is interrupted, its stale is saved in process control block. Then interrupt service routine is dispatched with the interrupt. The process state is restored and resumed. So, overheads include the cost of saving and restoring process state, restoring instruction into pipeline when process is restarted and cost of flushing the instruction pipeline

Nichamon Han-idhikul 5610545684

(14.5) Discuss the strengths and weaknesses of implementing an access matrix using access lists that are associated with objects

Access lists are fit for user needs. When a user creates an object, he can specify which domain can access the object and what operations are allowed. However, determing the set of access rights for each domain is difficult because access-right information for a particular domain is not localized.

(14.9) What is the need-to-know principle? Why is it important for a protection systen to adhere to this principle

Need-to-know principle

A process should be allowed to access only resources that it has authorization or it currently requires to complete its task. It is useful in limiting the amount of damage a faulty process can cause in the system.

(14.13) How does the principle of least privilege aid in the creation of protection system?

Least privilege principle dictates that programs users and systems be given just enough privileges to perform their tasks so that the failure does the minimum damage. OS provides mechanisms to enable privileges when they are needed and to disable them when they are not needed. This principle manages users by creating a separate account for each user