Main Memory

Address Binding

Logical Versus Physical Address Space

Swapping
Swapping on Mobile Systems

Contiguous Memory Allocation and Memory Protection

Memory Allocation

Fragmentation

Segmentation and Segmentation Hardware

Paging, Hardware Support and Protection

Translation look-aside buffer (TLB).

Address-space identifiers (ASIDs)

valid–invalid bit.

page-table length register (PTLR)
Hierarchical Paging

Consider a computer system with a 32-bit logical address and 4-KB page size. The system supports up to 512 MB of physical memory. How many entries are there in each of the following?

- A conventional single-level page table.

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

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

a. 3085
b. 42095
c. 215201
d. 650000
e. 2000001

Question 4

What is the purpose of paging the page tables?

Name two differences between logical and physical addresses.

Why are page sizes always powers of 2?

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.

Compare the main memory organization schemes of contiguous-memory allocation, pure segmentation, and pure paging with respect to the following issues: a. external fragmentation b. internal fragmentation c. ability to share code across processes

Virtual Memory

Virtual memory and virtual address space

Memory management unit (MMU)

Demand Paging

page fault and the procedure for handling this page fault

locality of reference

Copy-on-Write vfork() vs fork()

Question 1

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 pages faults would occur for the following replacement algorithms?

Remember that all frames are initially empty, so your first unique pages will cost one fault each

- LRU replacement
- FIFO replacement
- Optimal replacement

Question 2

Under what circumstances do page faults occur? Describe the actions taken by the operating system when a page fault occurs.

Question 3

Discuss the hardware support required to support demand paging?

Question 4

What is the copy-on-write feature, under what circumstances is its use beneficial? What hardware support is required to implement this feature?

Question 5

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?

Page Replacement
FIFO Page Replacement
Belady's anomaly:
Optimal Page Replacement
LRU Page Replacement

Thrashing
Cause of Thrashing

Mass-Storage Structure
Disk Scheduling
FCFS Scheduling
SSTF Scheduling
SCAN Scheduling
C-SCAN Scheduling
LOOK Scheduling
RAID Structure

Question 5
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:
2,069, 1,212, 2,296, 2,800, 544, 1,618, 356, 1,523, 4,965, 3681

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

b. SSTF
c. SCAN
e. C-SCAN

File –System Interface

File

File Attributes
File Operations

File locks provide functionality similar to reader–writer locks, covered in Section 5.7.2. A **shared lock** is akin to a reader lock in that several processes can acquire the lock concurrently. An **exclusive lock** behaves like a writer lock; only one process at a time can acquire such a lock.

Mandatory or **advisory** file-locking mechanisms.

File Types
File Structure
Access Methods
Sequential Access
Direct Access
Directory and Disk Structure
Storage Structure

Paths
Path names can be of two types: absolute and relative. An **absolute path name** begins at the root and follows a path down to the specified file, giving the directory names on the path. A **relative path name** defines a path from the current directory. For example, in the tree-structured file system of Figure 11.11, if the current directory is root/spell/mail, then the relative pathname prt/first refers to the same file as does the absolute path name root/spell/mail/prt/first.

File-System Mounting
File Sharing

File –System Implementation
File-System Structure
Layered file system.
File-System Implementation
Partitions and Mounting
Directory Implementation
Allocation Methods
Free-Space Management