First question

- A 32-bit virtual memory system has 8KB pages.
  - How many bits of the address are used by the *byte offset*?
  - How many bits of the address are used by the *page number*?
  - How *many pages* are there in a process address space?
Answer

- A 32-bit virtual memory system has 8KB pages.
- How many bits of the address are used by the byte offset?
  - \( \log_2 8K = 13 \) bits
- How many bits of the address are used by the page number?
  - \( 32 - 13 = 19 \) bits
- How many pages are there in a process address space?
  - \( 2^{19} = 512K \) pages
Second question

- Which of the following properties apply to these page replacement policies:
  - Handles fairly well real-time processes.
  - Has an unacceptably high overhead.
  - Occasions additional context switches.
  - Performs very poorly.
  - Used by a popular operating system.

- Policies include **Local LRU, Local FIFO, Global LRU, Global FIFO, BSD Clock, Mach, Windows**
Answer

- Which of the following properties apply to these page replacement policies:
  - Handles fairly well real-time processes.
    - Windows
  - Has an unacceptably high overhead.
    - Local LRU, Global LRU
  - Occasions additional context switches
    - BSD Clock, Mach, Windows
  - Performs very poorly.
    - Local FIFO, Global FIFO
  - Used by a popular operating system.
    - BSD Clock, Mach, Windows
Third question

- A 32-bit FFS file system has a block size of 8 kilobytes. How many bytes—not blocks—of a given file can be accessed:
  - Directly from the i-node?
  - With one level of indirection?
  - With two levels of indirection?
FFS organization (I)

Twelve direct blocks

\[ \begin{align*} b/4 \text{ block addresses} \\ b/4 \text{ indirect block addresses} \end{align*} \]

Block size \( b \geq 4\text{KB} \)

\[ b/4 \times b/4 \text{ double indirect blocks} \]
FFS organization (II)

- Twelve direct blocks
- 2,048 block addresses
- 2,048 indirect block addresses
- Up to 4M double indirect blocks but file size cannot exceed 4GB

Block size = 8KB
A 32-bit FFS file system has a block size of 8 kilobytes. How many bytes—not blocks—of a given file can be accessed:

- Directly from the i-node?
  - $12 \times 8K = 96K$ bytes

- With one level of indirection?
  - $(8K/4) \times 8K = 16M$ bytes

- With two levels of indirection?
  - $4G - 16M - 96K$ bytes
Fourth question

- Describe the contents of a UNIX directory entry.
Answer

- Describe the contents of a UNIX *directory entry*.

  - A UNIX directory entry contains a name and an i-node number.
Fourth question

Why does Berkeley FFS use blocking writes for all metadata updates?
Answer

Why does Berkeley FFS use *blocking writes* for all *metadata updates*?

- To guarantee the durability of these updates and the consistency of the file system in the presence of system crashes.
Fourth question

- How can we prevent deadlocks by denying the *hold and wait* condition?
Answer

- How can we prevent deadlocks by denying the *hold and wait* condition?

  - *We should impose an all-or-nothing resource allocation scheme that does not allow processes to hold resources while waiting for more resources.*
Fourth question

- What is the purpose of the *valid bit* in a virtual memory system?
Answer

- What is the purpose of the **valid bit** in a virtual memory system?

  - It tells whether the page table entry is valid and the page is in main memory or it is invalid and the page is missing.
Fourth question

- What does the Berkeley Fast File System do to reduce *disk seeks*?
Answer

- What does the Berkeley Fast File System do to reduce *disk seeks*?

  - It organizes each disk partition into cylinder groups that contain a fragment of the i-node table and enough data blocks to let most files reside in the same cylinder group as their i-node.
Fourth question

- What is the main advantage of mapped files?
Answer

- What is the main advantage of mapped files?

- Mapped files transfers the data blocks that a process accesses directly into the address space of the process, thus eliminating the context switches that would otherwise be required to bring them from the I/O buffer.
Fifth question

- What is the function of the /etc/group file in the UNIX file system?

- What would happen in an intruder could modify that file?
Answer

- What is the function of the /etc/group file in the UNIX file system?
  - The /etc/group contains a list of all protection groups on the system with their memberships.

- What would happen if an intruder could modify that file?
  - The intruder could add the account he—or she—is using to each group and gain group access to all files on the system.
Sixth question

- What is *internal fragmentation*? (5 points)

- Why is it a *more important* issue in *file systems* than in *virtual memory systems*?
Answer

- What is *internal fragmentation*?

  - Fragmentation occurs because each process must occupy a fixed number of pages and each file must occupy a fixed number of blocks. Hence the last file page and the last file block are almost never full.
Answer

- Why is it a more important issue in file systems than in virtual memory systems?

- Many file systems have numerous files whose size can be a fraction of the block size of the file system while even the smallest process will contain several pages.
Seventh question

- Explain how hashed page tables work.
Answer

Hash table

PN

hash

PFN

PN = page number
PFN = page frame number
The table only contains the pages that reside in main memory.
More explanations

- As hashed page tables only contain the pages that currently reside in main memory, they are accessed through a hash function.
- Each hash table entry contains a pointer to the head of a—possibly empty—linked list containing the page table entries corresponding to a given hashed value of their page number.