1. Consider a processor that supports virtual memory. It has a virtually indexed physically tagged cache, TLB, and page table in memory. Explain what happens in such a processor from the time the CPU generates a virtual address to the point where the referenced memory contents are available to the processor.

- CPU generates a virtual address(VA)
- CPU checks TLB(Translation Lookaside Buffer), a cache memory
    - If cache exists, return a physical address(PA)
    - If not, lookup a page table
- CPU check the page table in memory
- The page table translates VA to PA
    - If present, update TLB with mapping between VA and PA
    - If not, page fault
- Cache Access
    - Check the virtually indexed physically tagged (VIPT) cache.
    - If cache exists, fetch data from the cache
    - If not, access memory
- Access memory with PA
- Update cache
- Return data to CPU

2. Distinguish between segmentation and paging.

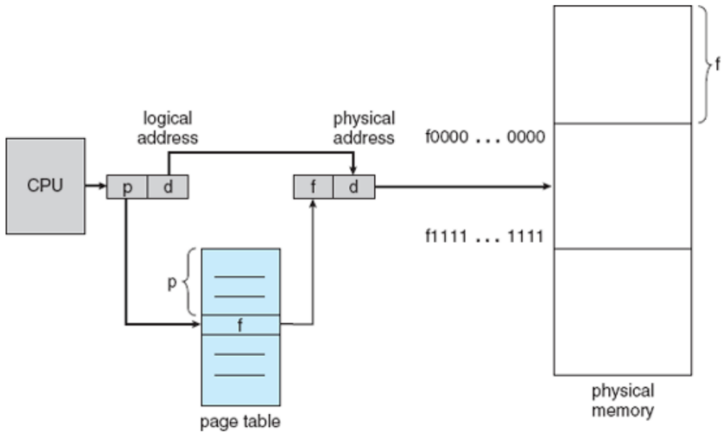


Figure 7.7. Paging hardware.

Galvin, P. B., Silberschatz, A., & Gagne, G. (2011). Operating System Concepts Essentials. Publisher. Figure 7.7.

Paging

- Divides memory into fixed-size pages.
- Uses a page table to map pages to frames in physical memory.
- Advantages: Eliminates external fragmentation, simplifies memory management.
- Disadvantages: May lead to internal fragmentation.

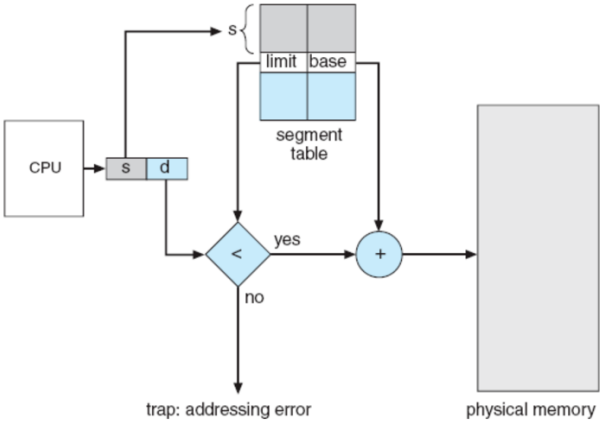


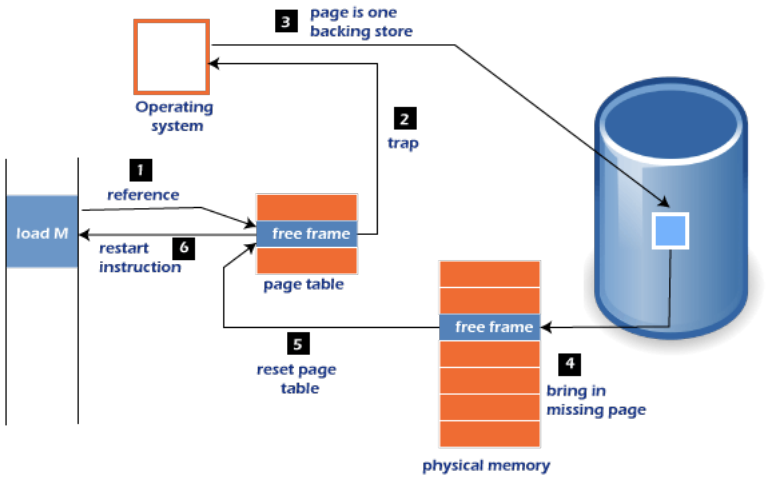
Figure 7.19. Segmentation hardware.

Galvin, P. B., Silberschatz, A., & Gagne, G. (2011). Operating System Concepts Essentials. Publisher. Figure 7.19.

Segmentation

- Divides memory into variable-sized segments.
- Each segment has a base address and limit.
- Advantages: Easier to manage logical divisions, more flexible.
- Disadvantages: Can lead to external fragmentation.

3. Explain all the actions from the time a process incurs a page fault to the time it resumes execution. Assume that this is the only runnable process in the entire system.



(Javatpoint, n.d.)

- Reference (**Step 1**): The CPU attempts to access a memory location, triggering a page fault if reference is invalid.
- Trap (**Step 2**): Trap to the operating system.
- Free Frame Search: The OS looks for a free frame in physical memory.
- Save process state
- Find free frame in a physical memory
- Page Replacement (if necessary): If no free frame is available, the OS selects a victim page to be replaced.
- Bring in Missing Page (**Step 4**): The OS reads the required page from the backing disk storage into physical memory.
- Update Page Table (**Step 5**): The OS updates the page table to reflect the new mapping of the virtual page to the physical frame.
- Restore the process state
- Restart Instruction (**Step 6**): The process resumes execution with the required page now in memory.
- Resume execution

4. Explain the following terms: working set of a process, thrashing, paging daemon, swapper, loader, and linker.

- Working set

The working set is the set of pages actively used by a process or in the most recent page references.

- Thrashing

A process spends more time on paging other than executing, so CPU utilization is going to be low.

- Paging daemon

A background process that manages page swapping.

- Swapper

A program responsible for swapping entire processes between memory and disk. Allocates memory and resolves symbolic references.

- Loader

The loader loads a executable program into memory for execution.

- Linker

The linker combines program components into a single executable file.

5. Explain page coloring and how it may be used in memory management by an operating system.

Page coloring is an optimization technique designed to increase cache utilization and improve CPU performance. Historically, processor caches were mapped directly to virtual memory, leading to significant issues such as:

- Cache Invalidation on Context Switches
- Data Aliasing Problems

To address these issues, modern systems now map caches directly to physical addresses, which introduced new challenges:

- Non-Contiguous Cache Mapping
- Cache Conflicts

Page coloring emerged as a solution to these challenges. This technique strategically assigns physical addresses to virtual addresses using an algorithm that prevents multiple virtual addresses from competing for the same cache lines. By optimizing cache management, page coloring enhances system performance and is implemented in operating systems like FreeBSD (Dillon, n.d.).

6. Explain clearly the costs associated with a process context switch.

A context switch is when the CPU switches from one process to another. This happens in a few steps. Each step can incur cost.

1. CPU Register State Save and Restore:
    - The current process's state must be saved into and restored from tis process Control Block (PCB). This is a time-consuming task and pure overhead. This incurs costs.
2. Memory Management
    - Page tables and the address space of the current process state must be preserved between switching. This process incurs costs such as memory copying and remapping.
3. Cache and TLB flushing
    - During context switching, the memory accessed by the new process may be different from the old process. In this case, the CPU cache or TLB needs to be invalidated or flushed. This invalidating or flushing costs performance degradation.
4. Scheduling and Synchronization Overheads
    - If the scheduling algorithm is complex, it can take more time to find the processes to change.
5. Interrupt Handling
    - Context switches are often caused by interrupts. This is a waste of resources. This increases the cost of context switches on interrupt-prone systems.
6. Overall System Complexity
    - The complexity of the system itself increases the size of data that needs to be stored, which in turn increases the overhead by requiring more work and time for the context switching process

Context switches are necessary for multitasking, but they incur overhead because the CPU must spend time storing and restoring process state instead of performing productive tasks. The cost of context switching depends on system’s hardware, the complexity of an operating system.

7. Explain the functionality of the different layers found in the OSI model.

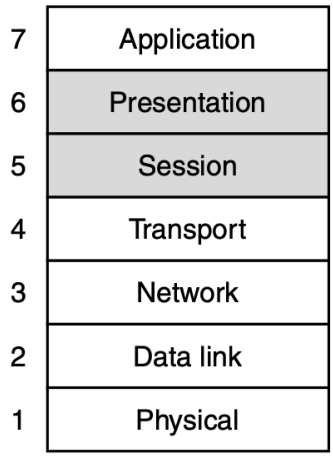


Figure 13.7 The OSI reference model is an abstract reference model developed by the standards body ISO.

Patt, Yale N., and Sanjay J. Patel. "Fundamentals of Networking and Network." Computer Systems: An Integrated Approach to Architecture and Operating Systems, Pearson, Year, pp. 631p.

7 layer - Application: User interfaces, services

This layer is responsible for providing network services directly to users with web browsers, email clients and file transfer applications. Users interacts with software applications over the network. Software applications implement the application layer protocols like HTTP, SMTP, FTP etc.

6 layer - Presentation: Data formatting, Encryption/ Decryption, compression

This layer plays a crucial role in data presentation and security. It formats data such as ASCII to EBCDIC. Also, it encrypts data before data transferred over network or decrypt data after it is transferred to another application. SSL is an example.

5 layer - Session: Connection management

The Session Layer manages communication sessions between applications. It is responsible for the establishment, maintenance, and termination of these sessions, ensuring that data exchanges are properly coordinated and synchronized. NetBIOS, RPC (Remote Procedure Call), and PPTP (Point-to-Point Tunneling Protocol) are examples of protocols in this layer.

4 layer - Transport: End-to-end communication, TCP/UDP

The transport layer is responsible for sending and receiving data from the application layer to and from the network. The TCP/UDP protocols are widely used protocols. TCP receives data, breaks it into chunks called segments/datagrams. It manages end-to-end communication between applications on different hosts. Port numbers are used to identify specific applications and services.

3 layer - Network: Routing, addressing

The network layer is responsible for routing packets from source to destination, separate from the transport layer. This separation allows for flexibility in network connections and routing paths. Key functions of the network layer include routing algorithms and service models. Routing algorithms determine the path for packets, while service models handle packet forwarding and switching. Routers implement network layer functionalities, consisting of physical, link, and network layers in their protocol stack. Internet Protocol(IP) is used to distinguish a device connected to a network.

2 layer - Data Link: Frames, error control

The link layer interfaces with hardware and is responsible for transmitting data over the physical medium. It deals with "frames" rather than packets, sometimes requiring multiple frames to transmit a single network layer packet. Link-layer protocols are categorized into random access (e.g., Ethernet) and taking turns (e.g., token ring) methods. Ethernet is currently the most popular link layer technology for connecting devices to local networks and the wider Internet. The Media Access and Control (MAC) is a subset of the link layer that handles access to the physical medium.

1 layer - Physical: Bit transmission

The physical layer indeed transfers bits physically over the network. It consists of cables, hubs, repeaters, NIC, connectors etc. It converts digital signals to analog signals. It deals with raw transmission of data.

References

Galvin, P. B., Silberschatz, A., & Gagne, G. (2011). Operating System Concepts Essentials. Publisher. Page number.

Javatpoint. (n.d.). *Page fault handling in operating system*. Retrieved August 28, 2024, from https://www.javatpoint.com/page-fault-handling-in-operating-system

Dillon, M. (n.d.). *Design elements of the FreeBSD VM system*. FreeBSD. https://docs.freebsd.org/en/articles/vm-design/#page-coloring-optimizations

Patt, Y. N., & Patel, S. J. (2010). Fundamentals of networking and network. In Computer systems: An integrated approach to architecture and operating systems (Chapter 13). Pearson.