T1 Exercises – Mehran Amiri

1. In fdisk, when using the p (print) command, there is a column labeled "sectors." What does "sector" refer to in this context?

The "sectors" column indicates the number of sectors a given partition occupies.
Sector numbers are used to define the start and end positions of partitions on the disk.
for example If a partition spans 204800 sectors and each sector is 512 bytes, the total size is 100 MB.

2. Partitioning recommendations for different server types:

Linux desktop systems:

- **/boot**: 512 MB–1 GB (for kernel and bootloader).
- **/ root**: 20–50 GB (for OS).
- **/opt** : 20GB (for applications)
- **/home**: Remaining space (for user data).
- **swap**: 1–2x RAM.

Linux servers for databases/web services with extensive logging:

- **/boot**: 512 MB–1 GB.
- **/**: 20–30 GB (OS and core services).
- **/var**: 50–100 GB (logs, databases, web content; separate for I/O isolation).
- **/tmp**: 10–20 GB (temporary files).
- **swap**: 1x RAM.

Linux servers in university labs/staging environments:

- **/boot**: 512 MB–1 GB.
- **/**: 20–30 GB.
- **/home**: Large partition (100 GB+) with quotas for user directories.

- **/tmp**: 10–20 GB (shared temporary storage).
- **swap**: 1–2x RAM

3. How does an OS run multiple applications with limited RAM?

Process scheduling: The OS uses a scheduler to allocate CPU time slices to processes based on priority and fairness, switching between them via context switches.

Loading process data into RAM: The OS loads only required portions (pages) of a process into RAM using **demand paging**, fetching data from disk as needed.

When RAM is insufficient: The OS moves less-used pages to **swap space** (disk) to free RAM for new processes, incurring performance penalties due to disk I/O.

Paging and swapping: **Paging** moves individual memory pages to disk; **swapping** moves entire processes. Both use disk as virtual memory when RAM is full

Memory management strategies:

- **Demand paging**: Loads pages only when accessed.
- **Page replacement**: Algorithms (e.g., LRU) decide which pages to swap out.
- **Virtual memory**: Maps process addresses to physical RAM or disk.

All pages loaded?: No, only active pages are loaded into RAM; others remain on disk until needed.

.

4. What is the Translation Lookaside Buffer (TLB), and its role in memory management?

TLB is a small, fast cache in the CPU that stores recent translations of virtual addresses to physical addresses. When the CPU accesses memory, it first checks the TLB. If the translation is found (TLB hit), access is fast. If not (TLB miss), the CPU performs a page table lookup.
the TLB improves performance for virtual memory systems, but a TLB miss (requiring page table lookup) can cause delays.

TLB reduces the overhead of virtual memory by avoiding repeated page table accesses, significantly improving memory access speed.

5. What are a page, a virtual page, and a context switch? Effects of swap size and page size?

Page: A fixed-size block of memory (e.g., 4 KB) used for virtual memory management.

Virtual page: A page in a process's virtual address space, mapped to physical RAM or disk.

Context switch: The process of saving one process's state and loading another's to switch CPU execution.

Increasing swap space:

- **Impact on context switching**: More swap allows more processes to run but slows context switches due to disk I/O for swapped pages.
- **Page size influence**: Larger pages (e.g., 2 MB) reduce TLB misses and page table overhead but increase I/O time when swapping, worsening context-switch performance if swap is heavily used.

6. What is a huge page?

A **huge page** is a larger memory page (e.g., 2 MB or 1 GB) compared to standard pages (4 KB).

- **Purpose**: Reduces page table size, lowers TLB miss rates, and improves memory access performance for large datasets.
- **When used**: In memory-intensive applications like databases (e.g., Oracle, PostgreSQL), HPC, or virtual machines, where large contiguous memory allocations are common.

7. What is memory fragmentation in RAM, and what problems can it cause?

Memory fragmentation occurs when free memory is scattered in small blocks across RAM, making it difficult to allocate large contiguous blocks.

Types:

  - **External fragmentation**: Free memory is scattered, causing allocation failures.

- **Internal fragmentation**: Allocated memory is larger than needed, wasting space.

- **Problems**:
  - Reduced memory utilization.
  - Allocation failures for large processes.
  - Performance degradation due to inefficient memory access or swapping.