Memory Management in Oracle Solaris: Paging and Virtual Memory

Kory Singleton & Kyle Clements

CSCI 431: Operating Systems - 11/4/25

Introduction to Paging

- Paging is a memory management scheme that breaks logical memory into fixed-size blocks called pages. It also divides physical memory into fixed-size blocks known as frames.
  - ➔ The main advantage of paging is that it allows a process's physical address space to be non-contiguous. This helps to avoid external fragmentation and makes memory allocation more flexible.
  - ➔ As a UNIX-based operating system, Oracle Solaris utilizes paging algorithms to handle memory management efficiently.

Virtual Memory Overview

- A memory management technique that abstracts the computer's physical RAM from individual processes. Each process is given the illusion that it has its own private, large, and continuous memory space.
- Oracle Solaris uses a demand-paged virtual memory system, meaning it only loads pages into physical memory when they are actually needed by a process.

<u>Core Benefits:</u>

  - Allows the system to run more applications than could fit into physical RAM simultaneously.
  - Keeps processes separate and secure from one another.
  - Maximizes the efficient use of physical RAM.

Solaris Virtual Memory Architecture

- The Solaris virtual memory system is structured in several key layers that work together to manage memory efficiently.
  - *Address Space Layer:* Manages the per-process virtual address spaces. Each process gets its own map of virtual memory.
  - *Page Cache Layer:* Handles file data that has been cached in memory.
  - *Physical Memory Layer:* Directly manages the physical page frames and is responsible for page replacement algorithms.

Page Tables and Translation

- A page table is the core map that translates a process's virtual addresses to physical RAM locations.
- Solaris uses HAT to manage address translation. This layer interacts with the hardware's Memory Management Unit.
- To accelerate the translation process, Oracle Solaris utilizes the CPU's Translation Lookaside Buffer, which is a high-speed cache for recently used page table entries.
    - ➔ When the CPU switches from one process to another, the TLB must be managed, often by flushing its entries to prevent one process from accessing another's memory.

Paging Process (Step-by-Step)

---

1. The Application requests memory
2. MMU (Memory Management Unit) references TLB (Transition Lookaside Buffer).
    1. If already mapped, the you’re done.
3. Check Page Table
4. If not found, Page fault occurs, so system has to get the address from the disk and find spot for it In the table.

Page Fault Handling in Oracle Solaris

- Solaris splits faults into 2 categories: minor and major faults.

Minor Fault

- Occurs when Page is already in memory but not mapped yet.
- Happens when a process forks and child gets copy-on-write mappings or Memory-mapped files are touched for the first time.
- “Appetizers at restaurant”

Major Fault

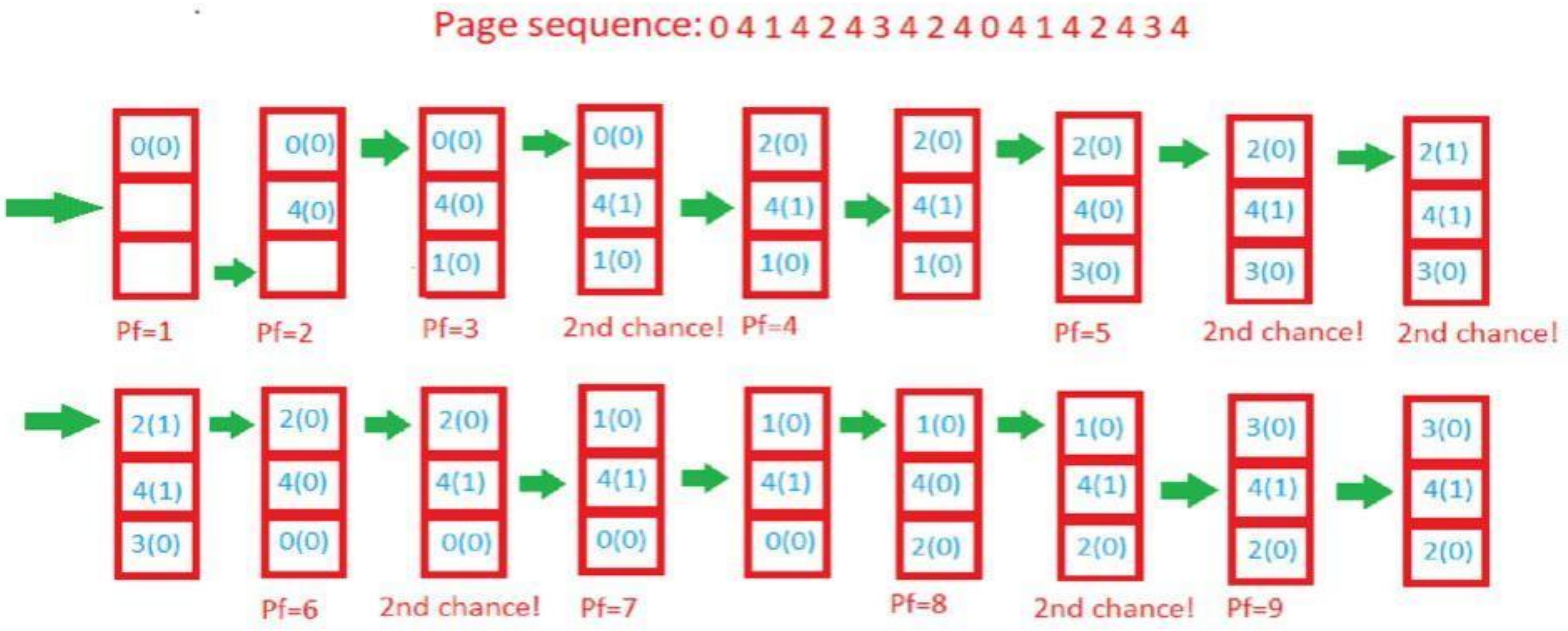
- Require gathering data from the disk.
- Data is already present in RAM, just needs to be pulled and updated into the data tables.
- “Entrees at Restaurant”

- Solaris uses 2 handlers for these: seg_vn for file-backed pages and anon_map for anonymous data.

Page Replacement Algorithms

- Solaris uses a modified version of the Clock Replacement algorithm (LRU).
- The System allows stored register with a 2nd chance bit if they are used before being replaced, allowing them to be “skipped” in the FIFO algorithm.

Example of Page Replacement Algorithm



Paging Daemons and System Control

- Daemons: a background process in Solaris that constantly watches and helps manage memory. Ultimate goal is prevent stalling or heavy slowing in the system
  - Pageout: Writes unused Pages to disc whenever free memory gets too low.
  - Fsflush: Cleans out cached file data to assure system stays responsive.
- There are also adjustable settings where administrators can fine-tune their performance.

Swap Space Management

- Swap Space: acts as the overflow area for virtual memory.
- When RAM fills up, Solaris uses Swap Space for additional memory.
    - Supports multiple swap devices which can alleviate load
- Lazy allocation prevent swap Space from being utilized unless it’s absolutely needed.
- Swap -s: Checks how much of swap Space is in use.
- Swap -l: switches the swap space

Summary / Takeaways

- Solaris paging: layered, demand-driven, efficient.
- Synergy between HAT, TLB, and pageout daemon.
- Benefits: reduced fragmentation, multitasking, scalability.
- “Oracle Solaris blends UNIX simplicity with intelligent memory management.”

Thank Your for Listening

- Questions/Concerns?