COS3 lektion 9

Nis Sarup

14. november 2010

9 Virtual-Memory Management

9.1 Background

- Most of Chapter 8 unnecessary with Virtual-Memory.
- Dynamic loading eases the restrictions.
- Having routines and data rarely used in a program take up physical memory is wasteful.
- Virtual-Memory allows programs to require much larger memory than what is available as physical memory.
- More programs can be run at one time, increasing CPU usage.
- Faster running of programs as less I/O is needed.
- Virtual-Memory separates logical memory from physical memory.
- Shared Libraries is easy with Virtual-Memory.
- Likewise is Shared Memory easy.

9.2 Demand Paging

- Demand Paging: Load pages only as they are needed.
- Similar to swapping, but with a lazy swapper.
- Valid/Invalid bit in page-table specifies wether a page is in memory or on disk.
- OS traps a request for an invalid page, loads the page into memory and restart the instruction.
- Demand Paging is not as quick as using only physical memory.
- The Effective Access Time depends on the speed of the medias and the probability of page faults.

9.3 Copy-on-Write

- Newly forked processes share pages.
- Pages that might diverge between the processes is marked for copy-on-write.
- As soon as one process write to one such page, it is copied to a new page and the written to.
- Processes will end up with as many shared pages as possible.

9.4 Page Replacement

- Page faults can also manifest ob over allocated systems, when no free pages are available.

9.4.1 Basic Page Replacement

- One solution is to find a frame not in use and replace it with the one needed by the process.
- The modify bit in a page table can be used to determine if a frame has been written to. If not, it does not need to be saved and the time to free a frame and load the requested frame is halved as the original frame does not need to be saved.

9.4.2 FIFO Page Replacement

- First In First Out: The oldest page is chosen for replacement.
- Belady's anomaly: For some page-replacement algorithms the page-fault rate may increase as the number of allocated frames increases.
- For FIFO the number of page faults at 4 frames is higher than at 3 fames.

9.4.3 Optimal Page Replacement

- Replace the page that will not be used for the longest period of time.
- Does not suffer from Belady's anomaly.
- It is difficult to implement though, as it requires future knowledge of which frame is going to be used.

9.4.4 LRU Page Replacement

- Least Recently Used algorithm.
- Does not suffer from Belady's anomaly.

9.5 Allocation of Frames

- Single-user system most simple.

9.5.1 Minimum number of Frames

- The minimum number of pages needed for a process to function depends on the computer architecture.
- Some instructions is more than two words big and therefor may need two pages.

9.5.2 Allocation Algorithms

- Easiest algorithm is to split m frames equally among n processes.
- Bigger processes probably need more frames than small processes.
- Proportional allocation takes this into account.

9.5.3 Global versus Local Allocation

- Global replacement: Victim frame is chosen among all frames.
- Local replacement: Victim frame is chosen only among the process' own frames.
- Higher priority processes might be able to choose victim frames from it's own frames or frames of lower priority processes.
- Global replacement results in greater system throughput.

9.5.4 Non-Uniform Memory Access

- Some parts of memory is faster to access than others.
- Memory allocation algorithms need to take NUMA into account for best performance.

9.6 Trashing

- A process is trashing if it spends more time paging than executing.
- Can be caused by the frames of the process getting below the number of frames required for executing set by the computer architecture.
- Trashing severely hampers performance.

9.6.1 Cause of Trashing

- If a system has low CPU utilization, the OS starts up additional processes.
- If a process is trashing, it uses little CPU.
- Adding processes only worsens the problem.
- Locality:
  - A set of frames a process needs to function in it's current state.
  - A process trashes if it has less frames than it's current locality needs.

9.6.2 Working-Set Model

- A number Δ is chosen.
- A list of unique frames used by a process in the last Δ-time is saved.
- The length of that list is a good approximation to how many frames it minimum needs to not trash.
- Doing a sum of that number, D, for all processes will give a number for frames currently needed.
- If D is larger than what is available, some processes must be stopped.
- Otherwise new processes can be started.

9.6.3 Page-Fault Frequency

- High PFF equals trashing.
- When PFF is low, start more processes.
- When PFF is high, suspend some processes.

9.7 Memory-Mapped Files

- Files and I/O can be mapped to virtual memory as well as physical memory.
- The first read of the file is as fast as usual.
- Subsequent reads read the file data directly from memory, greatly improving read-speed.
- This can lead to increased performance.

9.9 Other Considerations

9.9.1 Prepaging

- Prepaging attempts to prevent initial high level of paging when a process starts by paging in all the frames needed at on start.

9.9.2 Page Size

- Smaller pages leads to better memory utilization and longer access times.
- Bigger pages leads to smaller page tables and shorter access times.

9.9.3 TLB Reach

- How much of the virtual memory the TLB can map at one point.
- If the TLB is smaller than a process' working set, the process spends a lot of time looking up where to find its data in the slow page table.

9.9.4 Inverted Page Tables

- Inverted Page Tables does not hold information about the complete logical memory space.
- additional information is needed when page faults occur.

9.9.5 Program Structure

- Taking into account the paged structure of memory when programming/compiling/running program, performance can be increased.

9.9.6 I/O Interlock

- Lock Bits in a page table lock the frames so they cannot be replace by other processes.
- This is handy for I/O-buffers. For example when reading data from a disk-drive.