3Name: Cicelia Siu

CS 219 – Assignment #9

Purpose: Become familiar with caching and virtual memory implementation

Points: 100

Reading/References: Chapter 5

Assignment:

Answer the following questions:

1. If all misses are classified into one of three categories (compulsory, capacity, or conflict) which misses are likely to be reduced when: [9 pts, 3 pts each]
   1. The program is rewritten so as to require less memory?

**Compulsory Misses, possibly less capacity misses**

* 1. If the clock rate of the processor executing the program is increased?

**None**

* 1. If the associativity of the existing cache is increased?

**Conflict Misses**

1. What is the purpose of swapping ? [5 pts]

**A process must be loaded into memory in order to execute. Swapping is a basic memory allocation. If limited RAM, swap a processes to local storage.**

1. For a virtual memory environment:
   1. Is it necessary for all of the pages of a process to be in main memory while the process is executing? [3 pts]

**No. need to contain only the active portions.**

* 1. Must the pages of a process in main memory be contiguous? [3 pts]

**No.**

* 1. In what order are the the pages of a process in main memory? [3 pts]

**Yes. sequential (bottom to top).**

1. What is the purpose of a Translation Lookaside Buffer (TLB)? [5 pts]

**A small specialized cache that keeps track of recently used address mappings to avoid having to do a page table lookup.**

1. Consider a fixed partitioning scheme with equal-size partitions of 2 - 16 bytes and a total main memory size of 2 - 24 bytes. A process table is maintained that includes a pointer to a partition for each resident process. How many bits are required for the pointer?

[7 pts]

physical address:

24 - 16 = 8 bits

(16 bits from virtual address)

**8 bits needed**

1. Give two reasons why the page size in a virtual memory system should not be very small. [7 pts]
2. **Requires larger page table, uses more RAM**
3. **Likely to increase overhead due to addition page swapping**
4. Give one reason why the page size in a virtual memory system should not very large.

[7 pts]

1. **inefficient and wasteful use of primary storage**
2. Consider a processor with a 16-entry TLB that uses 2 KB pages. What are the performance consequences of this memory system if a program accesses at least 2 MB of memory at a time? Can anything be done to improve the performance? [10 pts]

**High miss rate in TLB (only 16x2 KB = 32 direct access). Increase TLB size to improve or page size increase.**

1. Consider a virtual memory system with the following properties:

● 36-bit virtual byte address

● 16 KB pages

● 32-bit physical byte address

What is the total size of the page table for each process on this processor, assuming that the valid, protection, dirty, and use bits take a total of 6 bits and that all the virtual pages are in use. Assume that disk addresses are not stored in the page table. Must sow work, final answer should be in megabytes1. [15 pts]

Virtual: (36 bit address)

|  |  |
| --- | --- |
| 22 | 14 |

Physical: (32 bit address)

|  |  |
| --- | --- |
| 18 | 14 |

222 x (6+18) = 100663296 bits = **12.582912 MB**

1. Compile (g++ -o progA progA.cpp), execute and time (time ./progA) each below programs. Explain, in detail, any difference in execution times and why. [10 pts]
2. Program A

#include <iostream>

#include <cstdlib>

using namespace std;

int main(int argc, char \*argv[])

{

int ROWS = 500000, COLS = 8192;

int cnt = 0, \*\*arr;

arr = new int\*[ROWS];

for (int r=0; r < ROWS; r++)

arr[r] = new int[COLS];

for (int r=0; r < ROWS; r++)

for (int c=0; c < COLS; c++)

arr[r][c] = cnt++;

return 0;

}

1. Program B

#include <iostream>

#include <cstdlib>

using namespace std;

int main(int argc, char \*argv[])

{

int ROWS = 500000, COLS = 8192;

int cnt = 0, \*\*arr;

arr = new int\*[ROWS];

for (int r=0; r < ROWS; r++)

arr[r] = new int[COLS];

for (int c=0; c < COLS; c++)

for (int r=0; r < ROWS; r++)

arr[r][c] = cnt++;

return 0;

}

Program A:

real 0m23.962s

user 0m8.188s

sys 0m5.257s

Program B:

real 4m15.619s

user 1m25.297s

sys 0m46.070s

**Program A is significantly faster than Program B, and is executed by row major. Program B experiences thrashing where pages are required to quickly swap in and back out.**

For conversions, refer to: http://www.matisse.net/bitcalc/

11) Complete the following table: [16 pts, 2 pts each entry]

|  |  |  |  |
| --- | --- | --- | --- |
| TLB | Page  Table | Data  Cache | Possible? Explain why or why not. |
| hit | hit | hit | Yes. Best w/ best performance |
| hit | hit | miss | Yes, but page table is neer really checked if TLB hits |
| miss | hit | hit | TLB misses, but entry found in page table, after retry, data is found in cache |
| miss | hit | miss | TLB misses, but entry found in page table, after retry, data misses in cache |
| miss | miss | miss | Yes, but TLB misses with a page fault, after retry, data must miss in cache |
| hit | miss | miss | No. cannot have a translation in TLB if page is not present in memory |
| hit | miss | hit | No. cannot have a translation in TLB if page is not present in memory |
| miss | miss | hit | No. Not possible to have a hit in data cache if there are misses in memory. |