95/100

COSC 450 Operating System Midterm #1

3/28/2024

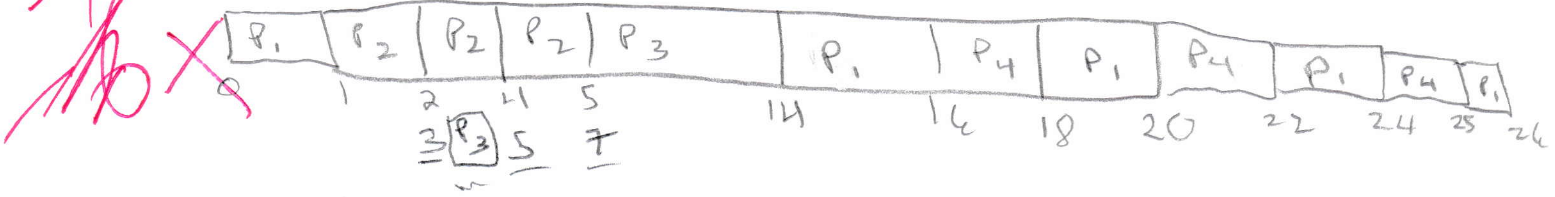
Name: Kyle Trofeglia.

1. (10 pt.) Consider the following set of processes (each processes are 100 % CPU-bounded).

Process	CPU-Time	Priorities	Arrival Time
P_1	8 7 5 3 1 0	1	0
P_2	4 3 1 0	3	1
P_3	9 — 0	3	2
P_4	5 3 1 0	1	3

Calculate average waiting time and average turnaround time with Preemptive Priority Queue process scheduling algorithms. With following condition

- High number has higher priority.
- Use Round-Robin between same priority with 2 units time quantum.
- Between processes with same priority, first come first serve.



average wait: $(((14-1)+(18-16)+(22-20)+(25-24))+0+(5-2)+((16-3)+(20-18)+(24-22)))/4 = 9.5$

Turnaround Time: $((26-0)+(5-1)+(14-2)+(25-3))/4 = 16$

2. (5 pt.) For a decimal virtual address 65350, compute the virtual page number and offset for a 2 KB and for a 4 KB page size in paging system.

- For 2KB:
  Page number: $65350 / 2 \times 2^{10} = \underset{\text{Truncate}}{31.91} \rightarrow \boxed{31}$ ✓

  Offset: $(65350 - 31 \cdot 2 \times 2^{10}) = \boxed{1862}$ ✓

- For 4KB:
  Page number: $65350 / 4 \times 2^{10} = 15.95 \rightarrow \boxed{15}$ ✓

  Offset: $(65350 - 15 \cdot 4 \times 2^{10}) = \boxed{3910}$ ✓

3. (10 pt.) Probabilistic model for multiprogramming.

   a) Lets *p* is an average fraction of time a process is waiting for I/O. If there are n processes in the memory at once, what will be the CPU utilization?

   p^n for n processes in memory at once, so CPU utilization = $\boxed{1-p^n}$ ✓

   b) A computer has 8 GB of memory, with operating system taking up 2GB and each program taking up 1 GB. With an 90 % average I/O wait, calculate CPU utilization

   $\frac{8\,GB - 2\,GB}{1\,GB/proc} = \frac{6\,GB}{1\,GB/proc} = 6\,processes$ $\quad p = 0.9 \quad \Rightarrow 1-(0.9)^6 = 0.47 \rightarrow \boxed{47\%}$ ✓

   c) By doubling the size of memory, what will be the CPU utilization?

   $\frac{16\,GB - 2\,GB}{1\,GB/proc} = 14\,processes$ $\quad p = 0.9 \quad \Rightarrow 1-(0.9)^{14} = 0.77 \rightarrow \boxed{77\%}$ ✓

4. (10 pt.) A Computer system generate a 64-bit virtual address use two-level page table. Virtual addresses are split into a 20-bit top-level page table field, a 20-bit second-level page table field and an offset.

   a) What is the size of each page ?

   Page Size $= 2^{64} / 2^{20} \cdot 2^{20} = 2^{24} \rightarrow 2^4 \cdot 2^{20} \rightarrow \boxed{16MB}$ ✓

   b) How many page tables are there?

   $\boxed{2^{20} + 1 \text{ Page tables}}$ ✓

   c) How many pages are there?

   $2^{20} \cdot 2^{20} = \boxed{2^{40} \text{ Pages}}$ ✓

   d) If system has 8GB memory, how many bits need to be reserved for saving page frame number in the each of page table entry?

   $8 \times 2^{30} / 2^{24} \rightarrow 2^{33}/2^{24} = 2^9$ Page Frames

   $\Downarrow$

   $\boxed{9 \text{ bits}}$ ✓

5. (10 pt.) Miss Computer presents a solution for the mutual exclusion with busy waiting. Variable **In** can be true or false. If **In** = **true,** a process is in busy waiting outside critical section. If **In** =false, a process set **In** =true and go to the critical section. Once a process finish its job in the critical section, set **In** = false, let other process enter critical section.

```
repeat
        while In ≠ false do
               ; (no-operation)
         In = true;
         ┌──────────────────┐
         │ Critical Section │
         └──────────────────┘
         In = false;
         ┌───────────────────┐
         │ Remainder Section │
         └───────────────────┘
until false
```

Does her method works for a mutual exclusion? Discuss Miss Wonder's solution. If her method guarantees the mutual exclusion, explain it how. If her method does not work for mutual, write a scenario lead to the situation which violates the mutual exclusion.

Miss Wonder's solution does not work for mutual exclusion.

Let there be two processes, P_1, P_2 with In = false. P_1 reads In = false. P_1 times out before entering critical section. P_2 now gets cpu time. P_2 reads In = false since P_1 did not yet set In to true. ✓ P_2 sets In = true and enters critical section. P_2 times out while in critical section. P_1 gets CPU time. Since it already read In = false, it sets In = true, which it already is, and enters critical section. Now P_1 and P_2 are in critical section which violates mutual exclusion.

6. (10 pt.) A computer can generate 32 bit virtual address for a process. This system has 4GB RAM and page size is 2KB. Following are mapping information for some pages getting from the page table
   - Page #1 is map to page from #5
   - Page #3 is map to page frame #1100
   - Page #5 is map to page frame #1230
   - Page #6 is map to page frame #2
   - Page #7 is map to page frame #95
   - Page #13 is map to page frame #7.
   - Page #14 is map to page frame #245
   - Page #15 is map to page frame #113

Calculate physical address for each of following virtual addresses

a) 29885 $/ 2\times2^{10} = 14.59 \rightarrow$ 14 maps to Page Frame # 245

Physical address $= 245 \cdot 2\times2^{10} + (29885 - 14 \cdot 2\times2^{10}) = \boxed{502973}$ ✓

b) 12530 $/ 2\times2^{0} = 6.12 \rightarrow$ 6 maps to Page Frame # 2

Physical address $= 2 \cdot 2\times2^{10} + (12530 - 6 \cdot 2\times2^{10}) = \boxed{4338}$ ✓

7. (10 pt.) A computer system generates a 32 bit virtual address for a process. This system has 32 GB RAM and page size is 4 KB.

a) If each entry in the page table needs 64 bits per entry, calculate the maximum possible size of the page table by Byte.

$\underbrace{2^{32}}_{\text{virtual address space}} / \underbrace{4\times2^{10}}_{\text{page size}} \Rightarrow 2^{32}/2^{12} = 2^{20}$ Pages

$2^{20} \cdot 64$bits $\Rightarrow 2^{20} \cdot 2^{6} = 2^{26}$ bits

$\Rightarrow 2^{26}/2^{3} = 2^{23}$ Bytes $\Rightarrow 2^{3}\times2^{20}$ Bytes $\Rightarrow \boxed{8\text{MB}}$ ✓

b) Page frame number information for each page must be saved in the page table. **How many bits** does it need to save page frame number information in a page table entry?

$\underbrace{32\times2^{30}}_{\text{memory size}} / \underbrace{4\times2^{10}}_{\text{Page size}} \Rightarrow 2^{5}\times2^{30} / 2^{2}\times2^{10} \rightarrow 2^{35}/2^{12}$

$\Rightarrow 2^{23}$ Page frames ✓ $\boxed{23 \text{ bits}}$

8. Short answer questions
   a) (1 pt.) Modern operating system has three types of schedulers: long term, short term and memory scheduler. Why the second generation operating system does not have short term scheduler? Batch system that sequentially runs magnetic tape does not support multiprogramming ✓
   b) (1 pt.) What are two main functions of OS for supporting multiprogramming? ✓
      a. Job Protection
      b. Job scheduling
   c) (1 pt.) What is race condition between processes? - multiple processes are trying to read/write from shared memory and the final result depends on what runs precisely when. ✓
   d) (1 pt.) Why effective short term scheduling algorithm is essential? ✓
      Context switching between processes has a lot of overhead ✓
   e) (1 pt.) A major problem with priority scheduling algorithm used in a scheduler is starvation. What is the solution of starvation?
      Aging ✓
   f) (1 pt.) What is the motivation of virtual memory?
      Big program sizes and limited amount of memory ✓
   g) (1 pt.) One of design issue of paging system is page size. Discuss disadvantages for smaller page size and bigger page size.
      smaller page size - Big page table ✓
      bigger page size - internal fragmentation, more wasted memory ✓
   h) (1 pt.) processes can use message queue or shared memory for inter-process communication. What is the main role of operating system for message queue and shared memory? OS creates each of these and is responsible for handling mutual exclusion and synchronization for message queue but not shared memory ✓
   i) (1 pt.) When a system uses virtual memory, virtual address do not directly onto the memory bus. Instead, it goes to an (MMU) that maps onto the physical memory address. ✓
   j) (1 pt.) Main goal of batch system for selecting proper scheduling algorithm are maximize throughput, minimize the turnaround time and CPU utilization. what are throughput and turnaround time?
      a. Throughput - # of processes completed ✓
      b. Turnaround time - Interval of time of submission to time of completion. (completion - scheduled start) ✓

9. (10 pt.) Mr. Computer tries to solve the race condition problem in the producer-consumer problem. He comes up with following solution. Are there any problems with his solution? If his method solves the race condition, show it. If his method could not solve the race condition, write a scenario lead to the race condition situation.

```
#define N 100
int count = 0;
void producer()
{
        int item
        while (ture)
        {
                item = produce_item();
                if (count == N)
                   sleep();
                insert_item(item)
                count = count + 1;
                if (count ==1)
                   wakeup(consumer);
        }
}
```

```
void consumer()
{
        int item;
        while(true)
        {
                if (count == 0)
                   sleep();
                item = remove_item();
                count = count – 1;
                if (count == N – 1)
                  wakeup(producer);
                consume_item(item);
        }
}
```

This method does not solve race condition. let count = N. Producer get CPU time. Count = N, Producer timeout before sleep. CPU time for consumer. Consumer consumes item count = N-1. Consumer CPU time end. Producer get CPU time. Producer already read count = N, so it goes to sleep. Consumer gets CPU time and consumes all until count = 0. Consumer sleep while wait for producer but producer can not be woken up. Consumer and producer both asleep forever. Deadlock

10. (5 pt.) Page table structure

a) What is the main motivation of multilevel page table?

X Reduce individual page table sizes, helps with searching pages with improved organization.

b) What is the main motivation of hashed page table?

Favors data allocation such that it benefits the speed of finding address by reducing overhead / failed search

c) How many entries are there in inverted page table?

A lot. There are no pages, so the entries is the number of page frames as that is what the entries in the table point to.

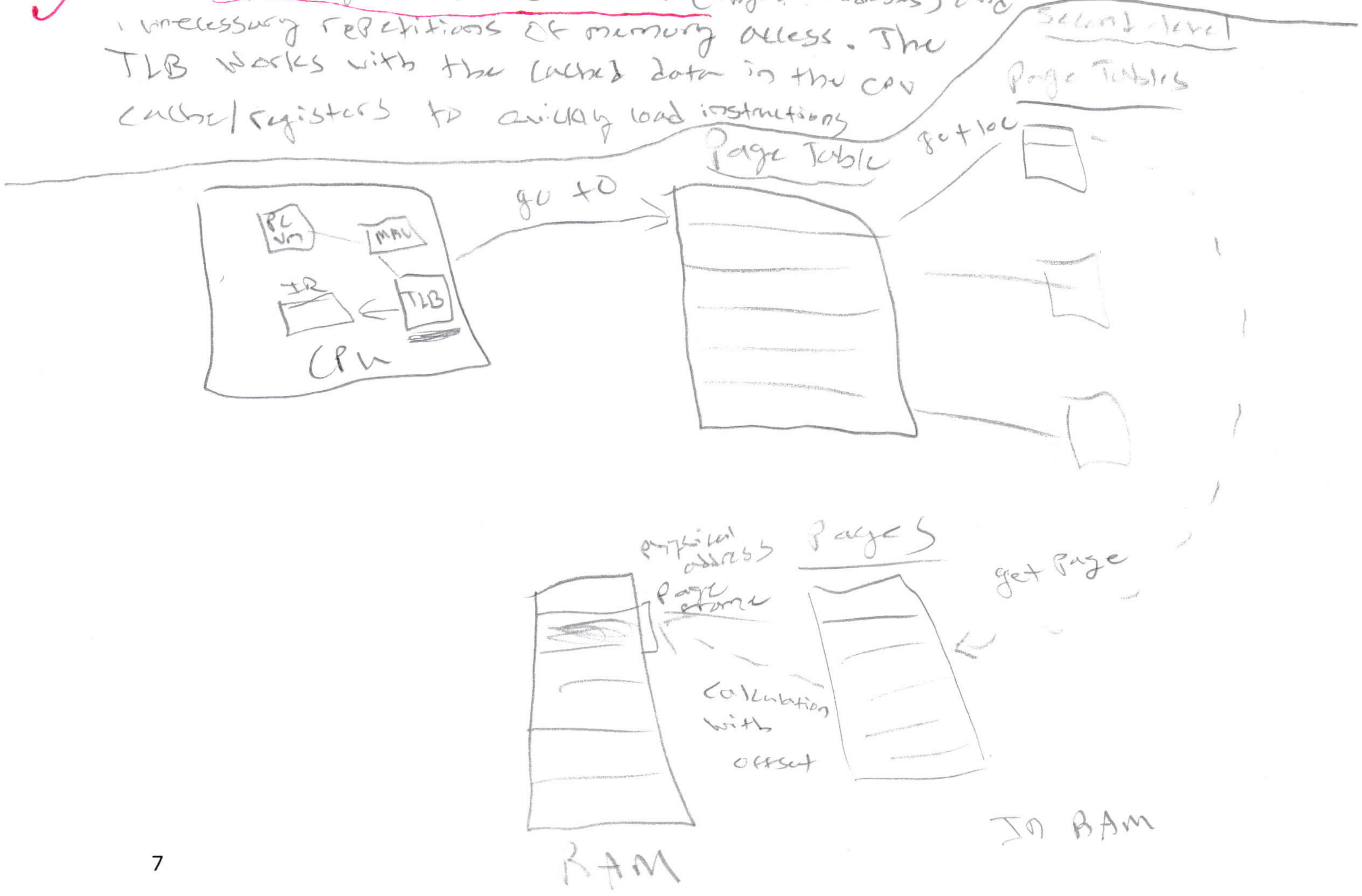
11. (10 pt.) Maintaining the page table in memory can potentially lead to slower memory access times. To access an instruction located in a virtual address, the system requires two memory access operations:

    a) Retrieving the page frame number from the page table stored in memory.
    b) Calculating the physical address by combining the page frame number with the offset.

    With hardware support such as a Translation Look-aside Buffer (TLB), the system can expedite memory access times. <u>Explain details how TLB support to speed up memory access time.</u>

Referencing the image below, the process to get a physical address is extensive and may be exhausting to repeatedly search virtual addresses and calculate physical addresses. The TLB speeds up the memory access time as it can bypass the memory access operations by temporarily caching recently accessed and frequently used instructions (Physical addresses) and unnecessary repetitions of memory access. The TLB works with the cached data in the CPU cache/registers to quickly load instructions

✓



Second level
Page Tables
Page Table
get loc
go to
PC
vn
MMU
IR
TLB
CPU
physical address
Pages
page frame
get page
calculation with offset
In RAM
RAM