Class Test 3 **Date: August 23, 2023**

BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY

Department of Computer Science and Engineering

January 2023 CSE 313: Operating System

Time: 30 minutes **Marks: 20**

Student Name: ______________________________ Student No: ____________

You are writing a 64-bit OS for a machine with 6GB of RAM (1GB = 2^{30}bytes). The OS must work for processes requiring up-to 200GB of memory. To virtualize the memory, you have decided to use *page tables* with 2KB pages. Now answer the following questions for this system.

1. What is the size of physical address (in bits)? (2)
   Ans. $\lceil \log_2(6 \times 2^{30}) \rceil = 33$ bit

2. What is the *minimum* size of virtual address (in bits)? (2)
   Ans. $\lceil \log_2(200 \times 2^{30}) \rceil = 38$ bit

3. What is the size of a page table entry (in bytes)? (2)
   Ans. $\lceil \frac{33-11+4}{8} \rceil = 4$ byte

4. How much memory will a single level page table use? (2)
   Ans. $4 \times 2^{38-11} = 2^{38-11+2} = 2^{29}$ byte $= 512$ MB

5. Design a multi level page table. Explain the translation using a diagram. (4)
   Ans.

   No. of entries in one page $= \frac{2^{11}}{4} = 2^{11-2} = 2^9$

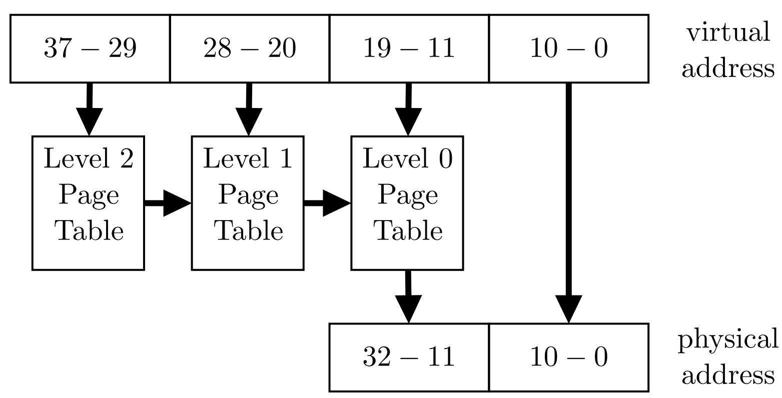

Figure 1: Multi level page table

6. The following program uses 1KB space for code, 4KB space for heap and 2KB space for stack. Here, a is an integer array allocated in heap and $sizeof(int) = 4$-bytes.

```
int sum = 0;
for (int i = 0; i < 1000; i++) {
    sum += a[i];
}
```

(a) How much memory will be used for this program by the multi level page table that you just designed? (4)

Ans. The program will use 3 pages for code and heap starting at $VA = 0$. and 1 page for stack ending at $VA = MAXVA = 2^{38} - 1$.

- No. of Level 2 page table = 1
- No. of Level 1 page table = 2 (one for code + heap, one for stack)
- No. of Level 0 page table = 2 (one for code + heap, one for stack)

So, total memory used by the page table = $(1 + 2 + 2) \times 2 = 10$ KB.

(b) Count the number of memory accesses that will be generated if TLB is available. (4)

- `sum, i` are in stack segment (or in register). They will generate $1 + 2 \times 1000 + 1001 = 3002$ (or 0) memory accesses and 1 (or 0) page table access (all other page table accesses will be to TLB).
- `a` is in heap segment. It will generate 1000 memory accesses and two (or three) page table accesses (all other page table accesses will be to TLB).
- instructions reside in code segment. Fetching instructions will generate $2 + 5 \times 1000 + 1 = 5003$ one (or two) page table access(es) and all other accesses will be to TLB.

So, total number of memory accesses (including page table access) = 9005 (or 6003) + 4 (or 3 or 5 or 6)