Homework 7

Chinese Name: _____

Pinyin Name: _____

Student ID: _____

E-Mail prefix: _____

10 1. **Put T (True) or F (False) for each of following statement. [2 points each]**

(a) ( ) The virtual memory system, via page table-based address translation mechanisms, enables multiple processes to securely share identical physical memory regions (e.g., shared code libraries) while maintaining isolation.

(b) ( ) A page-table walker (PTW) handles and resolves page faults by loading missing pages into memory.

(c) ( ) The size of the virtual address space accessible to the program cannot be larger than the size of the physical address space.

(d) ( ) Multi-level page tables optimize memory usage by storing complete physical addresses at the leaf level, enabling direct memory access without translation overhead.

(e) ( ) When a page fault occurs, the operating system must evict a resident page before loading the requested page into memory.

24 2. **TLB [24 points]**

The system uses a 32-bit virtual address space with byte-addressable memory and 30-bit physical addresses, employing a single-level page table with 4KB pages and a 4-way set-associative TLB (total 16 entries, LRU replacement policy).

8 (a) Answer the following questions. Specify bit ranges as [high:low] with zero-based indexing (bit 0 = LSB). [8 points]

The range of bits for the VPN (virtual page number) _____.

The range of bits for the offset within a page _____.

The range of bits for the TLB tag _____.

The range of bits for the TLB set number _____.

4 (b) Assume the TLB is initially empty, and the accessed VPNs are $4, 17, 20, 8, 21, 5, 17, 25, 29$ in sequence. Complete Final TLB State (Table 1). [4 points]

Table 1: Final TLB State

Index	VPN
0	
1	
2	
3	

12 (c) Based on part (b), subsequent accesses to VPNs are $12, 24, 21, 5, 4, 8, 17, 12, 29$. [12 points]

- For Table 2:
  - Mark TLB-hit VPNs with ✓
  - Mark VPNs that **evict others** with ◯
  - Mark VPNs that **are evicted** with ×
  - **If a newly annotated element has causal relationship with an existing element, both should be marked (e.g., when element A replaces element B, then A is marked with ◯ and B is marked with ×). Otherwise, only the current access is annotated.**
- Complete Final TLB State (Table 3)

Table 2: Access Sequence Change

VPN	Marks	VPN	Marks	VPN	Marks
4		17		5	
17		25		4	
20		29		8	
8		12		17	
21		24		12	
5		21		29	

Table 3: Final TLB State

Index	VPN
0	
1	
2	
3	

12 3. **Multi-level Page Tables [12 points]**

The system employs a 48-bit virtual address space with byte-addressable memory, using 4KB pages and 32-bit page table entries (PTEs) where 8 bits are reserved for protection and valid flags. The address translation uses a three-level page table hierarchy. Each of the three page table index fields is 10 bits.

4 (a) Given the virtual address `0x3456789ABC`. Suppose the contents of the page table entries of the three-level page table are as follows:

- The base address of the first-level page table is `0x1000`.
- The base address of the second-level page table is `0x2000`.
- The base address of the third-level page table is `0x3000`, and the PTE with index 3 points to the Physical Page Frame (PPN) `0x4000`. [4 points]

Final physical address:_____.

4 (b) Calculate the total size of all second-level page tables. [4 points]

4 (c) Consider an x86-64 system with 64-bit virtual addressing (48-bit usable), 4KB page tables, and 8-byte PTEs, supporting both 4KB pages (4-level: PML4→PDPT→PD→PT) and 2MB huge pages (3-level, no PT). When using 2MB huge pages for application data, how is the virtual address partitioned? **Specify bit allocations for PML4, PDPT, PD indices and page offset as [high:low] ranges using zero-based indexing**. [4 points]

28 4. **Page Faults [28 points]**

The system implements a single-level page table for 32-bit virtual addresses in byte-addressable memory, using 4KB pages with 32-bit page table entries (PTEs) that reserve 8 bits for protection and valid flags. Given the program segment:

```
1 int w[24][64];
2 ...
3 for ( i=0; i<24; i++ )
4    for ( j=0; j<64; j++ ) w[i][j]=10;
```

Assume the array w[24][64] (row-major order, starting at `0x392E00`, 4-byte integers) is initially not present in main memory, with no page replacement during execution.

12 (a) After execution: [12 points]

- The array spans _____ pages in the virtual address space.
- The number of page faults triggered is _____.
- Fault addresses (page base address): _____.

8 (b) If the page size is 128 bytes, and the array is stored in column-major order, after execution: [8 points]

- The number of page faults triggered is_____.

8 (c) Assume page size is 16KB with two-level page table. What is the maximum size of the physical memory that can be supported by this computer? [8 points]

26 5. **Page Table Walk [26 points]**

The system employs a 32-bit virtual address space with a three-level page table (2 bits per level for indexing), using 16B pages and 32-bit PTEs.

Assume:

- **Page table base register**: Set to `0`
- **Free PPNs list (allocated in order)**: `0x3`, `0x9`, `0x12`, `0x19`
- **Partial Memory Content**:Table 4 shows a snapshot of physical memory. Column `Cont.(PA)` indicates that the content stored at this location is a physical address (page base address).

Table 4: Partial Memory Content

PA	Cont. (PA)	PA	Cont. (PA)	PA	Cont. (PA)	PA	Cont. (PA)
0x00	0x10	0x28	0x70	0x50		0x78	
0x04	0x20	0x2C	0x110	0x54		0x7C	
0x08		0x30		0x58	0x160	0x80	0x200
0x0C		0x34		0x5C		0x84	0x240
0x10	0x40	0x38		0x60	0x130	0x88	
0x14		0x3C		0x64	0x150	0x8C	0x260
0x18		0x40		0x68	0x170	0x90	
0x1C		0x44		0x6C		0x94	
0x20		0x48	0x120	0x70	0x140	0x98	
0x24	0x50	0x4C		0x74		0x9C	

22 (a) Based on the content of Partial Memory Content (Table 4), for Process A's virtual address access sequence `0x162` and `0x324`, answer the following questions. All numerical answers should be in hexadecimal format, the leading zeros are optional, and **all values must represent the final state of the system**. [22 points]

Virtual Address `0x162`:

For virtual address `0x162`: VPN = _____, Page Offset = _____.

Using a three-level page table with two index bits per level:

The Level 1 index is _____, the Level 2 index is _____, and the Level 3 index is _____.

The initial value of the Page Table Base Register is `0x0`.

First, in the Level 1 page table, the corresponding entry is $PA =$ _____, $Cont. =$ _____. Thus, the Level 2 page table base address is _____.

Next, in the Level 2 page table, the corresponding entry is $PA =$ _____, $Cont. =$ _____. Thus, the Level 3 page table base address is _____.

Finally, in the Level 3 page table, the corresponding entry is $PA =$ _____, $Cont. =$ _____. Thus, the target page frame base address is _____.

Combining the page frame base address with the Page Offset, the final physical address is _____, resulting in a Page _____ (Hit/Miss).

Virtual Address `0x324`**:**
For virtual address `0x324`: VPN = _____, Page Offset = _____.
Using a three-level page table with two index bits per level:
The Level 1 index is _____, the Level 2 index is _____, and the Level 3 index is _____.

The initial value of the Page Table Base Register is `0x0`.

First, in the Level 1 page table, the corresponding entry is $PA =$ _____, $Cont. =$ _____. Thus, the Level 2 page table base address is _____.

Next, in the Level 2 page table, the corresponding entry is $PA =$ _____, $Cont. =$ _____. Thus, the Level 3 page table base address is _____.

Finally, in the Level 3 page table, the corresponding entry is $PA =$ _____, $Cont. =$ _____. Thus, the target page frame base address is _____.

Combining the page frame base address with the Page Offset, the final physical address is _____, resulting in a Page _____ (Hit/Miss).

4 (b) Calculating the size of the increase in the physical memory occupied by process A during the given sequence of virtual address accesses (counting only process-exclusive physical pages, excluding shared pages, kernel structures, and other unrelated components). [4 points]