Operating System (H)
Southern University of Science and Technology
Mengxuan Wu
12212006

Assignment 4

Mengxuan Wu

Question 1

During address translation, the CPU hardware is responsible for

1. Remember the base and bound values for currently running process.
2. Perform the translation of virtual address to physical address.
3. Raise an exception if the process tries to access an illegal address.
4. Support instructions to let the operating system change the base and bound registers, but not user mode programs.

The operating system is responsible for

- Allocating and de-allocating memory for processes.
- Remember the base and bound registers for each process, and load them into the CPU when the process is scheduled.
- Handling the exception raised by the CPU when a process tries to access an illegal address.

In general, the OS tells the CPU what the base and bound registers should be, and what handler should be called when an exception is raised. The CPU performs the actual address translation with the given base and bound registers, and raises an exception if the address is illegal.

Question 2

Segmentation	**Paging**
Size of chunks	
Size of chunks is variable. OS can allocate memory of different sizes to different chunks.	Size of chunks is fixed (e.g., page size).
Management of free space	
Free space is tracked by the OS. When a process requests memory, the OS finds a free	Free space is tracked by the OS. When a process requests memory, the OS finds one

chunk of the appropriate size (always contiguous) and gives it to the process.	or more free pages (could be non-contiguous) and gives them to the process.
Context switch overhead	
The OS needs to load the segmentation table for the new process and change corresponding registers.	The OS needs to change the base address register of page table for the new process and flush the TLB.
Fragmentation	
Segmentation can cause external fragmentation. If the OS cannot find a contiguous chunk of memory, it cannot allocate memory to the process.	Paging solves the problem of external fragmentation. The OS can allocate non-contiguous pages to a process.
Status bits	
Segmentation can have status bits to indicate growth direction, read/write/execute permissions.	Paging can have status bits to indicate read/write/execute permissions, valid/invalid pages (valid bit), dirty/clean pages (dirty bit), present/absent pages (present bit), accessed/not accessed pages (accessed bit).

Question 3

Three levels of page tables are required.
Since the page size is 8KB, the offset is 13 bits. For 46-bit virtual addresses, we need $46 - 13 = 33$ bits for the page number. Since each page table entry is 4 bytes, we can fit $8\text{KB}/4\text{B} = 2^{11}$ entries in each page table. Therefore, we need $33/11 = 3$ levels of page tables.
The format of the virtual address is as follows:

11 bits (Level 1)	11 bits (Level 2)	11 bits (Level 3)	13 bits (Offset)

Question 4

(a)

The page size is $2^{12} = 4\text{KB}$, since the offset is 12 bits.
Then $32 - 12 = 20$ bits are left for the page number. The maximum page table size is $2^{20} \times 4\text{B} = 4\text{MB}$.

(b)

For `0xC302C302`, the first-level page number is `0x30C` (10 bits), and the offset is `0x302` (12 bits).
For `0xEC6666AB`, the second-level page number is `0x266` (10 bits), and the offset is `0x6AB` (12 bits).

Question 5

```
[Step 1 allocation] start: 0, end: 3
[Step 2 allocation] start: 4, end: 7
[Step 4 allocation] start: 8, end: 15
[Step 5 allocation] start: 4, end: 5
```

For the first allocation, the buddy system allocates the initial 4 frames to the process. For the second allocation, the buddy system allocates 4 frames, following the first 4 frames. Then the second 4 frames are de-allocated. Now the whole memory is segmented into:

- 4 frames allocated to the process.
- 4 frames free.
- 8 frames free.
- 16 frames free.
- 32 frames free.

Thus, when requesting 8 frames, the buddy system will allocate the 8 frames starting from the 8th frame. And the memory is segmented into:

- 4 frames allocated to the process.
- 4 frames free.
- 8 frames allocated to the process.
- 16 frames free.
- 32 frames free.

Then, 2 frames are allocated. The buddy system splits the first 4 free frames into 2 frames and 2 frames. The memory is segmented into:

- 4 frames allocated to the process.
- 2 frames allocated to the process.
- 2 frames free.
- 8 frames allocated to the process.
- 16 frames free.
- 32 frames free.