CS302:

Assignment6 Report

Name：陆荻芸 SID：12011537

1 Answer of Question 1

In general, CPU hardware is responsible to redirect application memory references to their actual locations in memory and OS get involved at key points to set up the hardware to ensure that the correct translations take place.

- **CPU Hardware**: First of all CPU provides the extra bit to ensure the OS can run on the privileged **kernel mode**. Moreover, a part of CPU calls **Memory Management Unit (MMU)** is responsible for translations between virtual address and physical address. Part of the MMU is used to store the **base and bounds registers** where the address translation takes place. The base register is used to store the value of the memory where a program starts to run. The bounds registers ensures that such addresses in base registers are within the confines of the address space. Base and bounds registers help to check if the address is valid, then do the translation to the valid address.
- **OS**: OS needs to handle requirements in 3 aspects to enable a complete address translation process: memory management, base/ bounds management and exception handling. OS provides a **free list** to manage the available memory in order to allocate and reclaim memory from processes. And it is responsible for **saving and storing** the value of base and bounds registers each time when context switch happens. Finally, it should be able to stop offending process when **exception** occurs.

CPU provides detailed support for executions while OS sets up everything for processes to execute successfully. Together they cooperate to enable the address translation.

2 Answer of Question 2

The basic idea of segmentation is generalized base/ bounds. Thus for segmentation, the chunk size is various according to how much memory space a process needs. But for paging, each page and page-frame are of fixed size, usually 4KB. Segmentation manages free memory through a free list which links all different size of free chunks together. For paging, OS also keeps a free list of all free pages, but since pages are of the same size, it can be allocated and managed more easier. When context switch takes place, OS should save and restore the segment registers pairs for segmentation, which is a large cost. But for paging, only the page table should be saved and restored each time for context switch. Segmentation produces external fragmentation as it split the memory space with segments in different sizes. Fixed size pages of paging scheme can divide memory space fully without external fragmentation, but the unallocated space of each page will produce internal fragmentation. Segmentation uses

bits to indicate the segment's kind and the status of if a segment grow positively or not. Protection bits are also appended to indicate the user's authority on this segment. Paging has 8 status bits for each page table entry including a present bit, a reference bit, a dirty bit, a user bit and 4 bits indicating how the hardware caching works. There is also a protection bit in PTE, indicating whether the page could be read from, written to, or executed from.

3 Answer of Question 3

- 8KB = 2^{13} $Bytes$, so the offset should be 13 bits.
- To fit a page into the page table, the page table should at least have number of entries: $8KB \div 4\ Bytes = 2^{11}$
- Level of page tables: $(46 - 13) \div 11 = 3$

L1 index [45 : 35]	L2 index [34 : 24]	L3 index [23 : 13]	offset [12 : 0]

Table 1: Format of 46-bit virtual address in lecture

4 Answer of Question 4

(a) Page size: $2^{12}Bytes$ = 4KB, Maximum page table size: $2^{20} \times 4 = 4MB$

(b) **0xC302C302**: 1100 0011 0000 0010 1100 0011 0000 0010(2)
1-st level page number: 780; Offset: 770
0xEC6666AB:1111 1100 0110 0110 0110 0110 1010 1011(2)
2-nd level page number: 614; Offset: 1707