Trieu Huynh Ba Nguyen 000405980

1. a) First-fit:

- 212KB is put in 500KB partition
- 417KB is put in 600KB partition
- 112KB is put in 500KB – 212KB = 288KB partition
- 426KB must wait

1. b) Best-fit:

- 212KB is put in 300KB partition
- 417KB is put in 500KB partition
- 112KB is put in 200KB partition
- 426KB is put in 600KB partition

1. c) Worst-fit:

- 212KB is put in 600KB partition
- 417KB is put in 500KB partition
- 112KB is put in 388KB partition
- 426KB must wait

=> Best-fit algorithm is the best option.

2. a) If a memory reference takes 200 nanoseconds, a paged memory reference will take 400 nanoseconds: 200 nanoseconds to access the page table and 200 nanoseconds to access the value in memory.

2. b) If 75% of all page-table references are found in the TLBs, then 25% of the references will not be found in the TLBs and will require regular memory access to access the page table (200 nanoseconds to access the page and another 200 nanoseconds to access the value in memory). Therefore:

$Effective\ tive = 75\% \times 200 + 25\% \times (200 + 200) = 250$ (nanoseconds)

3. a) $219 + 430 = 649$

3. b) $2300 + 10 = 2310$

3. c) Invalid address, size of segment 2 is 100 and offset in logical address is 500.

3. d) $1327 + 400 = 1727$

3. e) Invalid address, size of segment 96 is 100 and offset in logical address is 112.

4. Segmentation and paging can be used in combination to enhance each other's benefits. The integration of these two techniques can offer improved performance and efficient memory management.

When dealing with a large page table, segmented paging can be useful. It involves collapsing an unused contiguous section of the page table into a single segment table entry with a page-table address of zero. This process can significantly reduce the size of the page table and improve the overall system's performance.

Paged segmentation is a suitable approach for dealing with long segments that require a considerable amount of time for allocation. By paging the segments, external fragmentation can be reduced, and memory allocation can be simplified. It eliminates the need to allocate large, contiguous blocks of memory and reduces the amount of wasted memory, which can be beneficial for systems with limited memory resources.

5. a)

$$Total\,number\,of\,pages = 32 = 2^5$$

$$Page\,size = 1024 = 2^{10}$$

$$\therefore LA = 2^{10} \times 2^5 = 2^{15}$$

Therefore, the number of bits required in logical address is 15 bits.

5. b)

$$Total\,number\,of\,frames = 16 = 2^4$$

$$Frame\,size = 1024 = 2^{10}$$

$$\therefore PA = 2^{10} \times 2^4 = 2^{14}$$

Therefore, the number of bits required in physical address is 14 bits.

6.

- Contiguous memory allocation scheme has the following characteristics:
  - It suffers from external fragmentation as address spaces are allocated contiguously and holes develop as old processes die and new processes are initiated.
  - It does not allow processes to share code since a process's virtual memory segment is not broken into non-contiguous fine-grained segments.
- Pure segmentation has the following characteristics:
  - It suffers from external fragmentation when a segment of a process is laid out contiguously in physical memory and fragmentation would occur as segments of dead processes are replaced by segments of new processes
  - It enables processes to share code.
- Pure paging has the following characteristics:
  - It suffers from internal fragmentation if a page is not completely utilized and will result in wastage of space.
  - It enables processes to share code at the granularity of pages.