Operating Systems

Assignment # 2: Solutions

[30 points]

1. 128 MB = 2^{27} Bytes

a. [2 points] Each segment is 64KB = 2^{16} bytes. The linked list has 2^{27} /2^{16} or 2^{11} nodes, each of 8 bytes, for a total of 2^{14} bytes.

b. [2 points] The bitmap needs 1 bit per allocation unit. With 2^{27} /n allocation units. Each allocation unit needs one bit in the bitmap. So we need (2^{27} /n) bits = 2^{24} /n bytes. For small n, the linked list is better.

c. [3 points] For large n, the bitmap is better. The crossover point can be calculated by equating these two formulas and solving for n. This is: 2^{24} /n = 2^{14}
The result is 1 KB. For n smaller than 1 KB, a linked list is better. For n larger than 1 KB, a bitmap is better.

Note: Of course, the assumption of segments and holes alternating every 64 KB is very unrealistic. Also, we need n <= 64 KB if the segments and holes are 64 KB.

2.

(a) [2 pts] A multilevel page table reduces the number of actual *pages of the page table* that need to be in memory at any point of time. For example, in a program with lots of instruction and data locality, we may only need the top-level page table (one page), one instruction page and one data page.

(b) [3 pts] The offset field requires 14 bits to address 16 KB. That leaves 24 bits for the page fields. Since each entry is 4 bytes, one page can hold 2^{12} page table entries and therefore requires 12 bits to index one page. So allocating 12 bits for each of the page fields will address all 2^{38} bytes.

---

3. [6 pts: 1 pt each]

- TLB: virtual address
- Page table: virtual address
- L1 cache memory: physical address
- L2 cache memory: physical address
- L3 cache memory: physical address

- The DRAM installed in your machine (i.e. your 16GB or 32GB of RAM): physical address.

---

4. [4 pts] Any one of the following is considered an advantage of the virtual memory. You need to cite four only.

- Gives process the illusion it has the whole memory for itself. This makes the life of the programmer easy.
- Makes linking much easier because the loader does not have to worry about other processes in the system.
- Makes loading much easier because the loader does not have to worry about other processes in the system.
- Protect processes from each other
- Gives the illusion of bigger memory than what is installed in the system.
- Makes it possible to share libraries among processes in dynamically linked libraries

---

5.

a. [2 pts] Only one TLB.
Because TLB is a piece of hardware that exists per CPU.

b. [2 pts] We will have two page tables.
Because the code created two processes. Each process needs its own page table.

c. [2 pts] We will have two TLBs.
There is a TLB per CPU, regardless of the number of processes.

d. [2 pts] We will still have two page tables.
The number of page tables depends on the number of processes and not the number of CPUs.

[Note: The information about cache memories is not needed to solve this problem].
