HW 1,2 Solution

1 HW 1

1. Ready list = [2 3 4] or [3 2 4], and semaphore waiting queue = [6 7] or [7 6].

2. If ready list = [2 3 4], then process 2 will be made current and process 1 is inserted into the ready list. The ready list changes to [3 1 4]. If ready list = [3 2 4], then process 3 becomes current and the list changes to [2 1 4].

3. send(5,170) checks the arguments, and places a message '170' to the process table entry representing process 5. Since process 5 is waiting for a message, the current process makes process 5 ready by calling ready(5). Process 5 is added to the ready list (just before process 4), and resched() is called. As a result, current process = 2 or 3 (whichever happened to be in the head of the ready list), ready list = [3 5 1 4] (if 2 was the first element), or [2 5 1 4].

4. Assume that semaphore waiting queue = [6 7]. signal(10) checks arguments, removes 6 from the queue and make process 6 ready by calling ready(6). Process 6 is placed in the ready list (just before process 4), and resched() is called. Either process 2 or 3 (whichever one that was in the head of the ready list) becomes current, and process 1 will be inserted into the ready list (just before process 4). As a result, current process = 2 or 3, ready list = [3 6 1 4] (or [2 6 1 4]), semaphore waiting list = [7].

   If semaphore waiting queue = [7 6], then current process = 7, ready list = [2 3 1 4] (or [3 2 1 4]), semaphore waiting queue = [6].

2 HW 2

1. First entry <4000, 600> changes to <4100, 500>. First 100 bytes allocated.

2. Fourth entry <5600, 800> changes to <5600, 200>. Last 600 bytes allocated.

3. Second entry <4800, 400> changes to <4700, 500>. Coalesced with a free block starting at address 4800.

4. Second entry <4800, 400> changes to <4800, 500>. Added 100 bytes at the end of the block.

5. Third entry <5400, 100> changes to <5400, 1000>. Remove fourth entry <5600, 800>. Coalesced with two adjacent blocks.