

Homework #9

Problem 1

- A. The program in the figure below has a bug. The thread is supposed to sleep for 1 second and then print a string. However, when we run it on our system, nothing prints. Why?
- B. You can fix this bug by replacing the exit function in line 9 with a Pthread function calls. What's it?

```
1  #include "csapp.h"
2  void *thread(void *vargp);
3
4  int main()
5  {
6      pthread_t tid;
7
8      Pthread_create(&tid, NULL, thread, NULL);
9      exit(0);
10 }
11
12 /* Thread routine */
13 void *thread(void *vargp)
14 {
15     Sleep(1);
16     printf("Hello, world!\n");
17     return NULL;
18 }
```

Problem 2

Using the progress graph in Figure 12.21, classify the following trajectories as either safe or unsafe. And draw them out.

- A. H2, L2, U2, H1, L1, S2, U1, S1, T1, T2
- B. H2, H1, L1, U1, S1, L2, T1, U2, S2, T2
- C. H1, L1, H2, L2, U2, S2, U1, S1, T1, T2

Problem 3

Refer to the I/O multiplexing based concurrent echo server in Figure 12.8, indicate the pros and cons of this event-driven programming approach based on I/O multiplexing.

Problem 4

Suppose this scenario, there's a lift with a max capacity of 18 persons at the same time (each person implemented as a thread). The only thing that a person will do in the lift is going up and down, which is indicated by the function "onLift()". Please fill in the code segment where is marked by "*Your code here*". (Use semaphore to guarantee the thread safety.)

```
#include "csapp.h"
void *thread(void *vargp);
unsigned int cnt;    /* count of persons */
sem_t sem;          /* semaphore */

int main()
{
    pthread_t tid1, ...;

    /* Initialize the semaphore. Your code here. */

    /* Create threads and wait */
    ...
    exit(0);
}

/* Thread routine */
void *thread(void *vargp)
{
    /* Your code here. */
}
```