# CS241 Lecture 13 Lawrence Angrave
# Working With pThreads. Introducing mutex locks

0. Can you call malloc from two threads?

Yes because it is "_____"

Which one of these is also safe to be used by two threads?
char *strerror(int errnum);
int strerror_r(int errnum, char *strerrbuf, size_t buflen);

1. Complete this code to print the thread id and an initial starting value. What does this code actually print? Why?

```
01  void* myfunc(void*ptr) {
02  printf("My thread id is %ld
            and I'm starting at %d\n",

 ____________________, ____________________);
03  return NULL;
04  }
05  int main() {
06  // Each thread gets a different value of i to
  process
07  pthread_t tid[10];
08  for(int i =0; i < 10; i++) {
09  pthread_create(& tid[i], 0, myfunc, &i);
10  }
11  //Two ways to wait for all threads to finish:
12
13
14
15
16
17
18
19
```

2 What is a critical section?

3 What is a mutex?

4 What are the two ways to create a mutex?

5 How do you lock and unlock a mutex?

6 When can you destroy a mutex and what is undefined behavior?

7. What does this code print? Will it always print the same output?

```
01  int counter;
02  void*myfunc2(void*param) {
03   int i=0; // stack variable
04   for(; i < 1000000;i++) counter ++;
05   return NULL;
06  }
07  int main() {
08   pthread_create(&tid1, 0, myfunc2, NULL);
09   pthread_create(&tid1, 0, myfunc2, NULL);
10   pthread_join(tid1,NULL);
11   pthread_join(tid2,NULL);
12   printf("%d\n", counter );
13  }
```

8 Use heap memory to pass starting information to each thread. Create two threads. Each thread will do half the work. The first thread will process 0..numitems/2 in the array. The second thread will process the remaining items. Any gotchas?

```
01  typedef struct work_ {
02
03
04  } work_t;

05  void calc (int * data, size_t nitems) {
06    size_t half = numitems/2;
07
08
09
10
11
12
13
14
15
16    pthread_create(&tid1, 0, imagecalc,____);
17  }
18  // Gotcha odd number of numitems. Memory leak?
```

9. Case study: Parallelize *AngraveCoin* miner

```
void search(long start, long end) {
  printf("Searching from 0x%lx to 0x%lx\n", start , end);
  for(long i = start; i <end; i++) {
    char message[100];
    sprintf(message,"AngraveCoin:%lx", i);

    unsigned char *res;

    res = SHA256(message, strlen(message), NULL);
    int iscoin;
    iscoin = (res[0] == 0)&&(res[1] == 0)&&(res[2] == 0);

    if(iscoin)
        printf("%lx %02x %02x %02x '%s'\n", i, res[0],
res[1], res[2] , message);
  }
  printf("Finished %lx to %lx\n", start, end);
}


long array[] = {0L, 1L <<25, 1L <<27, 1L <<33};
int main() {
  search(array[0], array[1]);
  search(array[1], array[2]);
  search(array[2], array[3]);
  search(array[3], array[4]);
  return 0;
}
```