CSC 369
Midterm review

Exercises and solutions will be posted online, so **don’t grab more than one copy** per group please (:

Start thinking about question 1 and we will go over the solutions soon

```
int first = -1;
int second = -1;
int got_one = 0;

void *do_exchange(void *arg){
  int value = *(int *)arg;
  lock_acquire(entry);

  if(!got_one) {

    got_one = 1;
    first = value;
    cv_wait(got_first, &entry);
    *(int *)arg = second;

  } else {
    got_one = 0;
    second = value;
    cv_signal(got_first)
    *(int *)arg = first;
  }

  lock_release(entry);
  fprintf(stderr,"%d -> %d\n",
          value, *(int *)arg);
          int first = -1;
          int second = -1;
          int got_one = 0;

          void *do_exchange(void *arg){
            int value = *(int *)arg;
T3   T2  T1T0lock_acquire(entry);

            if(!got_one) {

              got_one = 1;
              first = value;
              cv_wait(got_first, &entry);
              *(int *)arg = second;

            } else {
              got_one = 0;
              second = value;
              cv_signal(got_first)
              *(int *)arg = first;
            }

            lock_release(entry);
            fprintf(stderr,"%d -> %d\n",
                    value, *(int *)arg);
```

```
           int first = -1;
           int second = -1;
           int got_one = 0;

           void *do_exchange(void *arg){
            int value = *(int *)arg;
T3   T2  T1 lock_acquire(entry);

            if(!got_one) {

             got_one = 1;
             first = value;
          T0 cv_wait(got_first, &entry);
             *(int *)arg = second;

            } else {
             got_one = 0;
             second = value;
             cv_signal(got_first)
             *(int *)arg = first;
            }

            lock_release(entry);
            fprintf(stderr,"%d -> %d\n",
                    value, *(int *)arg);
```

```
int first = -1;
int second = -1;
int got_one = 0;

void *do_exchange(void *arg){
  int value = *(int *)arg;
```

T3 T2

```
  lock_acquire(entry);

  if(!got_one) {

    got_one = 1;
    first = value;
```

T0

```
    cv_wait(got_first, &entry);
    *(int *)arg = second;

  } else {
    got_one = 0;
```

Second = 1

```
    second = value;
    cv_signal(got_first)
    *(int *)arg = first;
  }

  lock_release(entry);
  fprintf(stderr,"%d -> %d\n",
          value, *(int *)arg);
```

T1

```
int first = -1;
int second = -1;
int got_one = 0;

void *do_exchange(void *arg){
  int value = *(int *)arg;
  lock_acquire(entry);

  if(!got_one) {

    got_one = 1;
    first = value;
    cv_wait(got_first, &entry);
    *(int *)arg = second;

  } else {
    got_one = 0;
    second = value;
    cv_signal(got_first)
    *(int *)arg = first;
  }

  lock_release(entry);
  fprintf(stderr,"%d -> %d\n",
          value, *(int *)arg);
```

T3 T2

T2 T0

Second = 1

T1

```
int first = -1;
int second = -1;
int got_one = 0;

void *do_exchange(void *arg){
  int value = *(int *)arg;
  lock_acquire(entry);

  if(!got_one) {

    got_one = 1;
    first = value;
    cv_wait(got_first, &entry);
    *(int *)arg = second;

  } else {
    got_one = 0;
    second = value;
    cv_signal(got_first)
    *(int *)arg = first;
  }

  lock_release(entry);
  fprintf(stderr,"%d -> %d\n",
          value, *(int *)arg);
```

T3 T2

T2 T0

Second = 3

T3

T1

```
int first = -1;
int second = -1;
int got_one = 0;
int stage = 0;   cv new_exchange;
void *do_exchange(void *arg){
  int value = *(int *)arg;
  lock_acquire(entry);

  if(!got_one) {

    got_one = 1;
    first = value;
    cv_wait(got_first, &entry);
    *(int *)arg = second;

  } else {
    got_one = 0;
    second = value;
    cv_signal(got_first)
    *(int *)arg = first;
  }

  lock_release(entry);
  fprintf(stderr,"%d -> %d\n",
          value, *(int *)arg);
```

```
int first = -1;
int second = -1;
int got_one = 0;
int stage = 0;
void *do_exchange(void *arg){
  int value = *(int *)arg;
  lock_acquire(entry);

  if(!got_one) {
    stage++;
    got_one = 1;
    first = value;
    cv_wait(got_first, &entry);
    *(int *)arg = second;
    stage = 0;
  } else {stage++;
    got_one = 0;
    second = value;
    cv_signal(got_first)
    *(int *)arg = first;
  }

  lock_release(entry);
  fprintf(stderr,"%d -> %d\n",
          value, *(int *)arg);
```

```
int first = -1;
int second = -1;
int got_one = 0;
int stage = 0;   cv new_exchange;
void *do_exchange(void *arg){
  int value = *(int *)arg;
  lock_acquire(entry);
      while (stage == 2)
          cv_wait(new_exchange, &entry);

  if(!got_one) {
    stage++;
    got_one = 1;
    first = value;
    cv_wait(got_first, &entry);
    *(int *)arg = second;
    stage = 0;
  } else {stage++;
    got_one = 0;
    second = value;
    cv_signal(got_first)
    *(int *)arg = first;
  }

  lock_release(entry);
  fprintf(stderr,"%d -> %d\n",
          value, *(int *)arg);
```

```
int first = -1;
int second = -1;
int got_one = 0;
int stage = 0;   cv new_exchange;
void *do_exchange(void *arg){
  int value = *(int *)arg;
  lock_acquire(entry);
while (stage == 2)
      cv_wait(new_exchange, &entry);
  if(!got_one) {
    stage++;
    got_one = 1;
    first = value;
    cv_wait(got_first, &entry);
    *(int *)arg = second;
    stage = 0;  cv_broadcast(new_exchange);
  } else {stage++;
    got_one = 0;
    second = value;
    cv_signal(got_first)
    *(int *)arg = first;
  }

  lock_release(entry);
  fprintf(stderr,"%d -> %d\n",
          value, *(int *)arg);
```