# CS241 LAB

Midterm lab review

- Worksheet (please look at questions on the sheet
- quizizz
- Code review
- Coding Trick & Style workshop
- Fill up midterm review form

- Worksheet
- quizizz
- Code review
- Coding Trick & Style workshop
- Fill up midterm review form

```
P1 Program 1 int counter;
```

#### Wrong

#### Good day!

```
static void * foo(void * tn) {
                                                          Thread 2
                                      Thread 1
                                                                             Counter
  int i;
  for (i = 0; i < 100000; i++)
    counter++;
                                                                             Counter = 1000
                                      Thread 1 read 1000
  return NULL;
                                                                             Counter = 1000
                                      Thread 1 1000 + 1
int main() {
  int i, N = 5;
                                                                             Counter = 1001
                                      Thread 1 write back 1001
  pthread t t[N];
                                                                             Counter = 1001
  for (i = 0; i < N; i++)
                                                            Thread 2 read 1001
    pthread create(&t[i],
NULL, foo, NULL);
                                                                             Counter = 1001
                                                            Thread 2 1001 + 1
  for (i = 0; i < N; i++)
    pthread join(t[i], NULL);
                                                            Thread 2 write back
                                                                             Counter = 1002
                                                            1002
  printf("%d\n", counter);
  return 0;
```

```
P1 Program 1 int counter;
```

#### Bad day!

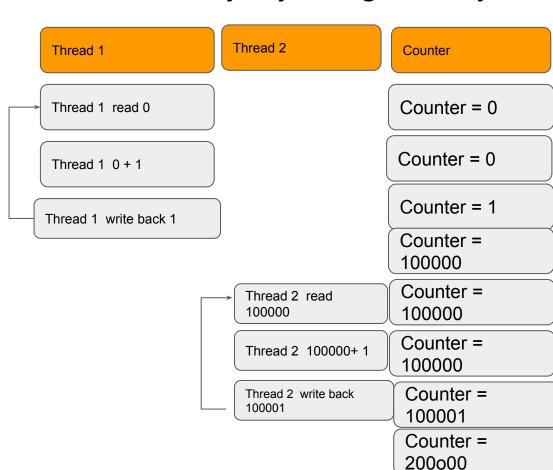
```
static void * foo(void * tn) {
                                                          Thread 2
                                      Thread 1
                                                                             Counter
  int i;
  for (i = 0; i < 100000; i++)
    counter++;
                                                                             Counter = 1000
                                      Thread 1 read 1000
  return NULL;
                                                                            Counter = 1000
                                      Thread 1 1000 + 1
int main() {
  int i, N = 5;
                                                            Thread 2 read 1000
                                                                             Counter = 1000
  pthread t t[N];
                                                                             Counter = 1000
                                                            Thread 2 1000 + 1
  for (i = 0; i < N; i++)
    pthread create(&t[i],
NULL, foo, NULL);
                                       Thread 1 write back 1001
                                                                             Counter = 1001
  for (i = 0; i < N; i++)
    pthread join(t[i], NULL);
                                                            Thread 2 write back
                                                                             Counter = 1001
                                                            1001
  printf("%d\n", counter);
  return 0;
```

Program 2
int counter;

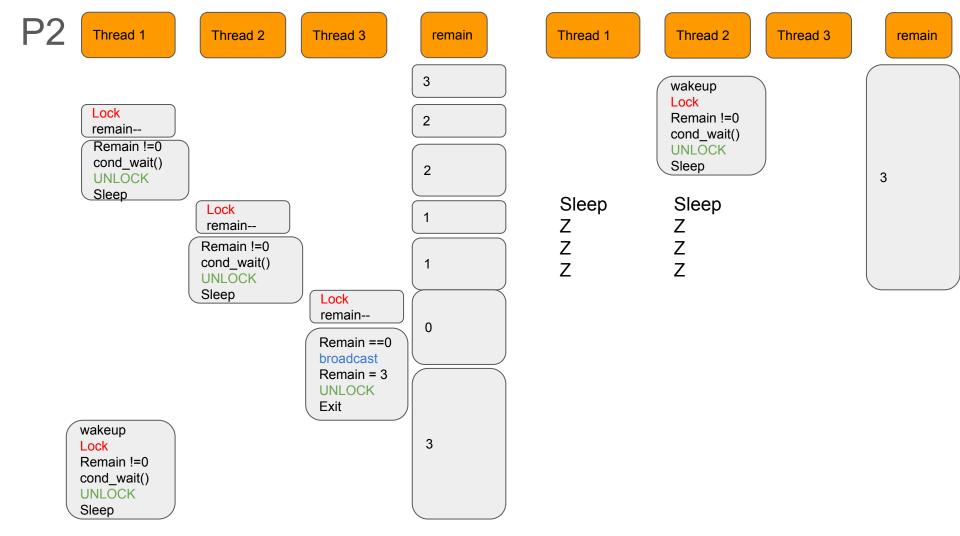
### Right

#### Everyday is a good day!

```
static void * foo(void * tn) {
  int i;
  for (i = 0; i < 100000; i++)
    counter++;
  return NULL;
int main() {
  int i, N = 5;
 pthread t t[N];
  for (i = 0; i < N; i++) {
    pthread create (&t[i], NULL,
foo, NULL);
    pthread join(t[i], NULL);
  printf("%d\n", counter);
  return 0;
```



Wrong Thread 1 Thread 2 Thread 3 remain pthread mutex lock(&m); remain--; 3 if (remain == 0) { Lock pthread cond broadcast (&cv); remain-remain = num threads; Remain !=0 cond wait() **UNLOCK** else { Sleep while(remain != 0) { Lock remain-pthread cond wait (&cv, &m); Remain !=0 cond wait() **UNLOCK** Sleep pthread\_mutex unlock(&m); Lock Sleep Lock remain--Remain ==0 broadcast wakeup wakeup Remain = 3 UNLOCK Exit wakeup Lock Remain!=0 cond wait() UNLOCK Sleep



```
main
                   Right
Program A
                                            Stack
void *foo(void *varqp) {
                                            ptr
  int myid;
  myid = *((int *)vargp);
  free (varqp);
  printf("Thread %d\n", myid);
int main() {
  pthread t tid[2];
  int i, *ptr;
  for (i = 0; i < 2; i++) {
    ptr = malloc(sizeof(int));
    *ptr = i;
    pthread create(&tid[i], 0, foo, ptr)
  pthread join(tid[0], 0);
  pthread join(tid[1], 0);
                                            heap
```

```
main
                   Right
Program A
                                             Stack
void *foo(void *vargp) {
                                             ptr
  int myid;
  myid = *((int *)vargp);
                                              thread1
  free (varqp);
                                             vargp
  printf("Thread %d\n", myid);
int main() {
  pthread t tid[2];
  int i, *ptr;
  for (i = 0; i < 2; i++) {
    ptr = malloc(sizeof(int));
    *ptr = i;
    pthread create(&tid[i], 0, foo, ptr)
  pthread join(tid[0], 0);
  pthread join(tid[1], 0);
                                             heap
```

```
main
                    Right
Program A
                                              Stack
void *foo(void *vargp) {
                                              ptr
  int myid;
 myid = *((int *)vargp);
                                              thread1
  free (varqp);
                                              vargp
  printf("Thread %d\n", myid);
                                              Myid = 0
int main() {
  pthread t tid[2];
  int i, *ptr;
  for (i = 0; i < 2; i++) {
    ptr = malloc(sizeof(int));
    *ptr = i;
    pthread create(&tid[i], 0, foo, ptr)
  pthread join(tid[0], 0);
  pthread join(tid[1], 0);
                                              heap
```

```
main
                    Right
Program A
                                             Stack
void *foo(void *vargp) {
                                             ptr
  int myid;
  myid = *((int *)vargp);
                                              thread1
  free (varqp);
                                             vargp
  printf("Thread %d\n", myid);
                                             Myid = 0
int main() {
  pthread t tid[2];
  int i, *ptr;
  for (i = 0; i < 2; i++) {
    ptr = malloc(sizeof(int));
    *ptr = i;
    pthread create(&tid[i], 0, foo, ptr)
  pthread join(tid[0], 0);
  pthread join(tid[1], 0);
                                             heap
```

```
main
                    Right
Program A
                                              Stack
void *foo(void *vargp) {
                                              ptr
  int myid;
  myid = *((int *)vargp);
                                               thread1
  free (varqp);
                                              vargp
  printf("Thread %d\n", myid);
                                              Myid = 0
                                               thread2
int main() {
                                              vargp
                                              Myid = 1
  pthread t tid[2];
  int i, *ptr;
  for (i = 0; i < 2; i++) {
    ptr = malloc(sizeof(int));
    *ptr = i;
    pthread create(&tid[i], 0, foo, ptr)
  pthread join(tid[0], 0);
  pthread join(tid[1], 0);
                                              heap
```

```
main
                    Right
Program A
                                              Stack
void *foo(void *vargp) {
                                              ptr
  int myid;
  myid = *((int *)vargp);
                                               thread1
  free (varqp);
                                              vargp
  printf("Thread %d\n", myid);
                                              Myid = 0
                                               thread2
int main() {
                                              vargp
                                              Myid = 1
  pthread t tid[2];
  int i, *ptr;
  for (i = 0; i < 2; i++) {
    ptr = malloc(sizeof(int));
    *ptr = i;
    pthread create(&tid[i], 0, foo, ptr)
  pthread join(tid[0], 0);
  pthread join(tid[1], 0);
                                              heap
```

```
main
                    Right
Program A
                                               Stack
void *foo(void *vargp) {
                                               ptr
  int myid;
  myid = *((int *)vargp);
                                                thread1
  free (varqp);
                                               vargp
                                                           "Thread 0"
  printf("Thread %d\n", myid);
                                               Myid = 0
                                               thread2
                                                           "Thread 1"
int main() {
                                               vargp
                                               Myid = 1
  pthread t tid[2];
  int i, *ptr;
  for (i = 0; i < 2; i++) {
    ptr = malloc(sizeof(int));
    *ptr = i;
    pthread create(&tid[i], 0, foo, ptr)
  pthread join(tid[0], 0);
  pthread join(tid[1], 0);
                                               heap
```

#### Wrong

```
Program B
```

```
void *foo(void *vargp) {
  int myid;
 myid = *((int *)vargp);
 printf("Thread %d\n", myid);
int main() {
 pthread t tid[2];
  int i;
  for (i = 0; i < 2; i++) {
   pthread create (&tid[i], NULL, foo, &i);
  pthread join(tid[0], NULL);
  pthread join(tid[1], NULL);
```

main

Stack i = 0

### Wrong

#### **Program B**

```
void *foo(void *varqp) {
 int myid;
 myid = *((int *)vargp);
 printf("Thread %d\n", myid);
int main() {
 pthread t tid[2];
 int i;
  for (i = 0; i < 2; i++) {
   pthread create (&tid[i], NULL, foo, &i);
 pthread join(tid[0], NULL);
 pthread join(tid[1], NULL);
```

main Stack i = 0thread1 vargp heap

### Wrong

#### Program B

```
void *foo(void *varqp) {
 int myid;
 myid = *((int *)vargp);
 printf("Thread %d\n", myid);
int main() {
 pthread t tid[2];
 int i;
  for (i = 0; i < 2; i++) {
   pthread create (&tid[i], NULL, foo, &i);
 pthread join(tid[0], NULL);
 pthread join(tid[1], NULL);
```

main Stack i = 1thread1 vargp heap

### Wrong

#### **Program B**

```
void *foo(void *varqp) {
 int myid;
 myid = *((int *)vargp);
 printf("Thread %d\n", myid);
int main() {
 pthread t tid[2];
 int i;
  for (i = 0; i < 2; i++) {
   pthread create(&tid[i], NULL, foo, &i);
 pthread join(tid[0], NULL);
 pthread join(tid[1], NULL);
```

main Stack i = 1thread1 Vargp Myid = 1heap

#### Wrong

#### **Program B**

```
void *foo(void *varqp) {
 int myid;
 myid = *((int *)vargp);
 printf("Thread %d\n", myid);
int main() {
 pthread t tid[2];
 int i;
  for (i = 0; i < 2; i++) {
   pthread create (&tid[i], NULL, foo, &i);
 pthread join(tid[0], NULL);
 pthread join(tid[1], NULL);
```

main Stack i = 1thread1 Vargp Myid = 1thread2 Vargp heap

### Wrong

#### **Program B**

```
void *foo(void *varqp) {
 int myid;
 myid = *((int *)vargp);
 printf("Thread %d\n", myid);
int main() {
 pthread t tid[2];
 int i;
  for (i = 0; i < 2; i++) {
   pthread create (&tid[i], NULL, foo, &i);
 pthread join(tid[0], NULL);
 pthread join(tid[1], NULL);
```

main Stack i = 2thread1 Vargp Myid = 1thread2 Vargp heap

#### Wrong

#### **Program B**

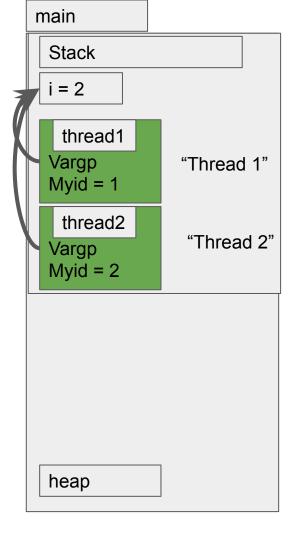
```
void *foo(void *vargp) {
  int myid;
 myid = *((int *)vargp);
 printf("Thread %d\n", myid);
int main() {
 pthread t tid[2];
  int i;
  for (i = 0; i < 2; i++) {
    pthread create(&tid[i], NULL, foo, &i);
  pthread join(tid[0], NULL);
  pthread join(tid[1], NULL);
```

main Stack i = 2thread1 Vargp Myid = 1thread2 Vargp Myid = 2heap

#### Wrong

#### **Program B**

```
void *foo(void *varqp) {
 int myid;
 myid = *((int *)vargp);
 printf("Thread %d\n", myid);
int main() {
 pthread t tid[2];
 int i;
  for (i = 0; i < 2; i++) {
   pthread create (&tid[i], NULL, foo, &i);
 pthread join(tid[0], NULL);
 pthread join(tid[1], NULL);
```



## Right

```
Program C
```

```
void *foo(void *vargp) {
  int myid;
  myid = (int) vargp;
  printf("Thread %d\n", myid);
int main() {
  pthread t tid[2];
  int i;
  for (i = 0; i < 2; i++)
    pthread create(&tid[i], 0, foo, i);
  pthread join(tid[0], 0);
  pthread join(tid[1], 0);
```

main

Stack

i = 0

```
Program C
```

```
void *foo(void *vargp) {
  int myid;
  myid = (int) vargp;
  printf("Thread %d\n", myid);
int main() {
  pthread t tid[2];
  int i;
  for (i = 0; i < 2; i++)
    pthread create(&tid[i], 0, foo, i);
  pthread join(tid[0], 0);
  pthread join(tid[1], 0);
```

main

Stack i = 0thread1 Vargp = 0

### Right

```
Program C
```

```
void *foo(void *vargp) {
  int myid;
  myid = (int) vargp;
  printf("Thread %d\n", myid);
int main() {
  pthread t tid[2];
  int i;
  for (i = 0; i < 2; i++)
    pthread create(&tid[i], 0, foo, i);
  pthread join(tid[0], 0);
  pthread join(tid[1], 0);
```

main

Stack
i = 1
thread1
Vargp = 0

### Right

```
Program C
```

```
void *foo(void *vargp) {
  int myid;
  myid = (int) vargp;
 printf("Thread %d\n", myid);
int main() {
  pthread t tid[2];
  int i;
  for (i = 0; i < 2; i++)
    pthread create(&tid[i], 0, foo, i);
  pthread join(tid[0], 0);
  pthread join(tid[1], 0);
```

main

Stack

i = 1

thread1
Vargp = 0
Myid = 0

### Right

#### **Program C**

```
void *foo(void *vargp) {
  int myid;
  myid = (int) vargp;
 printf("Thread %d\n", myid);
int main() {
  pthread t tid[2];
  int i;
  for (i = 0; i < 2; i++)
    pthread create(&tid[i], 0, foo, i);
  pthread join(tid[0], 0);
  pthread join(tid[1], 0);
```

main Stack i = 1thread1 Vargp = 0"Thread 0" Myid = 0thread1 "Thread 1" Varqp = 1Myid = 1heap

```
Program D
```

```
sem t s; /* Semaphore s */
void *foo(void *vargp) {
  int myid;
  myid = *((int *)vargp);
  sem post (&s);
  printf("Thread %d\n", myid);
int main() {
  pthread t tid[2];
  int i;
  sem init(&s, 0, 0); /* S=0 INITIALLY */
  for (i = 0; i < 2; i++) {
    pthread create(&tid[i], 0, foo, &i);
    sem wait(&s);
  pthread join(tid[0], 0);
  pthread join(tid[1], 0);
```

main

Stack

i = 0

Sema = 0

thread1 Vargp

```
Program D
```

```
sem t s; /* Semaphore s */
void *foo(void *vargp) {
  int myid;
  myid = *((int *)vargp);
  sem post (&s);
  printf("Thread %d\n", myid);
int main() {
  pthread t tid[2];
  int i;
  sem init(&s, 0, 0); /* S=0 INITIALLY */
  for (i = 0; i < 2; i++) {
    pthread create(&tid[i], 0, foo, &i);
    sem wait(&s);
  pthread join(tid[0], 0);
  pthread join(tid[1], 0);
```

main

Stack

i = 0

Sema = 0

Sem\_wait

thread1 Vargp

```
Program D
```

```
sem t s; /* Semaphore s */
void *foo(void *vargp) {
  int myid;
  myid = *((int *)vargp);
  sem post (&s);
  printf("Thread %d\n", myid);
int main() {
  pthread t tid[2];
  int i;
  sem init(&s, 0, 0); /* S=0 INITIALLY */
  for (i = 0; i < 2; i++) {
    pthread create(&tid[i], 0, foo, &i);
    sem wait(&s);
  pthread join(tid[0], 0);
  pthread join(tid[1], 0);
```

main

Stack

i = 0

thread1

Sem\_wait X Vargp
Myid = 0

Sema = 0

```
Program D
```

```
sem t s; /* Semaphore s */
void *foo(void *vargp) {
  int myid;
  myid = *((int *)vargp);
  sem post (&s);
  printf("Thread %d\n", myid);
int main() {
  pthread t tid[2];
  int i;
  sem init(&s, 0, 0); /* S=0 INITIALLY */
  for (i = 0; i < 2; i++) {
    pthread create(&tid[i], 0, foo, &i);
    sem wait(&s);
  pthread join(tid[0], 0);
  pthread join(tid[1], 0);
```

Sema = 0

Sem wait

Sem post

main

Stack

i = 0

thread1

Vargp Myid = 0

```
Program D
```

```
sem t s; /* Semaphore s */
void *foo(void *vargp) {
  int myid;
  myid = *((int *)vargp);
  sem post (&s);
  printf("Thread %d\n", myid);
int main() {
  pthread t tid[2];
  int i;
  sem init(&s, 0, 0); /* S=0 INITIALLY */
  for (i = 0; i < 2; i++) {
    pthread create(&tid[i], 0, foo, &i);
    sem wait(&s);
  pthread join(tid[0], 0);
  pthread join(tid[1], 0);
```

a b t

Sema = 1

Sem\_wait √

main

Stack

i = 1

thread1

Vargp Myid = 0

```
Program D
```

```
sem t s; /* Semaphore s */
void *foo(void *vargp) {
  int myid;
  myid = *((int *)vargp);
 sem post(&s);
  printf("Thread %d\n", myid);
int main() {
  pthread t tid[2];
  int i;
  sem init(&s, 0, 0); /* S=0 INITIALLY */
  for (i = 0; i < 2; i++) {
    pthread create(&tid[i], 0, foo, &i);
    sem wait(&s);
  pthread join(tid[0], 0);
  pthread join(tid[1], 0);
```

main

Stack

i = 1

Sema = 0

Sem\_wait

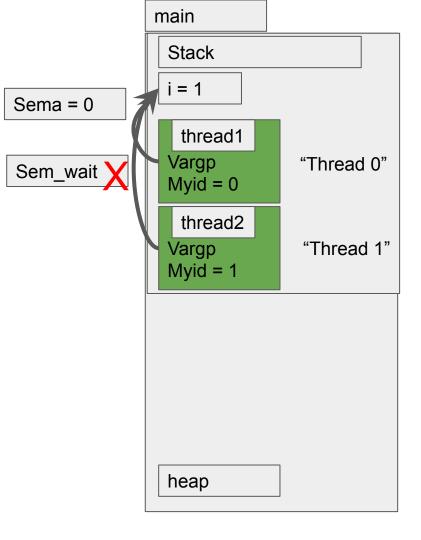
thread1

Vargp Myid = 0

thread2
Vargp
Myid = 1

```
Program D
```

```
sem t s; /* Semaphore s */
void *foo(void *vargp) {
  int myid;
  myid = *((int *)vargp);
  sem post (&s);
  printf("Thread %d\n", myid);
int main() {
  pthread t tid[2];
  int i;
  sem init(&s, 0, 0); /* S=0 INITIALLY */
  for (i = 0; i < 2; i++) {
    pthread create(&tid[i], 0, foo, &i);
    sem wait(&s);
  pthread join(tid[0], 0);
  pthread join(tid[1], 0);
```



#### **WRONG**

#### **Program E**

```
sem t s; /* Semaphore s */
void *foo(void *vargp) {
 int myid;
  sem wait(&s);
 myid = *((int *)vargp);
 sem post(&s);
 printf("Thread %d\n", myid);
int main() {
 pthread t tid[2];
 int i;
  sem init(&s, 0, 1); /* S=1 INITIALLY */
  for (i = 0; i < 2; i++) {
   pthread create(&tid[i], 0, foo, &i);
 pthread join(tid[0], 0);
 pthread join(tid[1], 0);
```

main

Stack

Sema = 1 i = 0

Sem wait v

thread1

Vargp

### Program E

```
sem t s; /* Semaphore s */
void *foo(void *vargp) {
  int myid;
  sem wait(&s);
  myid = *((int *)vargp);
  sem post(&s);
  printf("Thread %d\n", myid);
int main() {
  pthread t tid[2];
  int i;
  sem init(&s, 0, 1); /* S=1 INITIALLY */
  for (i = 0; i < 2; i++) {
    pthread create(&tid[i], 0, foo, &i);
  pthread join(tid[0], 0);
  pthread join(tid[1], 0);
```

main Stack i = 1Sema = 1thread1 Sem wait V Vargp thread2 Vargp heap

## Program E

```
sem t s; /* Semaphore s */
void *foo(void *vargp) {
 int myid;
  sem wait(&s);
 myid = *((int *)vargp);
  sem post(&s);
 printf("Thread %d\n", myid);
int main() {
 pthread t tid[2];
 int i;
  sem init(&s, 0, 1); /* S=1 INITIALLY */
  for (i = 0; i < 2; i++) {
   pthread create(&tid[i], 0, foo, &i);
 pthread join(tid[0], 0);
  pthread join(tid[1], 0);
```

main Stack i = 1thread1 Vargp Myid = 1thread2 Vargp

Sema = 0

Sem\_wait

heap

### Program E

```
sem t s; /* Semaphore s */
void *foo(void *vargp) {
  int myid;
  sem wait(&s);
  myid = *((int *)vargp);
  sem post(&s);
  printf("Thread %d\n", myid);
int main() {
  pthread t tid[2];
  int i;
  sem init(&s, 0, 1); /* S=1 INITIALLY */
  for (i = 0; i < 2; i++) {
   pthread_create(&tid[i], 0, foo, &i);
  pthread join(tid[0], 0);
  pthread join(tid[1], 0);
```

main Stack i = 2Sema = 0thread1 Vargp Sem wait Myid = 1thread2 Vargp heap

#### Program E

```
sem t s; /* Semaphore s */
void *foo(void *vargp) {
 int myid;
  sem wait(&s);
 myid = *((int *)vargp);
  sem post(&s);
 printf("Thread %d\n", myid);
int main() {
 pthread t tid[2];
 int i;
  sem init(&s, 0, 1); /* S=1 INITIALLY */
  for (i = 0; i < 2; i++) {
   pthread create(&tid[i], 0, foo, &i);
 pthread join(tid[0], 0);
  pthread join(tid[1], 0);
```

main Stack i = 2Sema = 1thread1 Vargp Sem wait V Myid = 1thread2 Vargp heap

## Program E

```
sem t s; /* Semaphore s */
void *foo(void *vargp) {
 int myid;
  sem wait(&s);
 myid = *((int *)vargp);
 sem post(&s);
 printf("Thread %d\n", myid);
int main() {
 pthread t tid[2];
 int i;
  sem init(&s, 0, 1); /* S=1 INITIALLY */
  for (i = 0; i < 2; i++) {
   pthread create(&tid[i], 0, foo, &i);
  pthread join(tid[0], 0);
  pthread join(tid[1], 0);
```

main

Stack

Sema = 0 i = 2

Sem wait X

thread1

Vargp Myid = 1

thread2

Vargp Myid = 2

heap

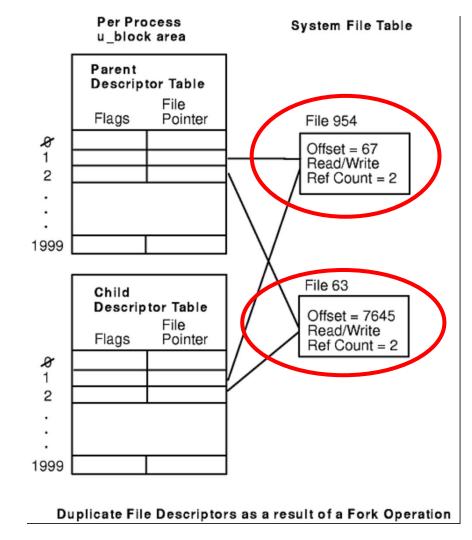
### Program E

```
sem t s; /* Semaphore s */
void *foo(void *vargp) {
 int myid;
  sem wait(&s);
 myid = *((int *)vargp);
 sem post(&s);
  printf("Thread %d\n", myid);
int main() {
 pthread t tid[2];
 int i;
  sem init(&s, 0, 1); /* S=1 INITIALLY */
  for (i = 0; i < 2; i++) {
   pthread create(&tid[i], 0, foo, &i);
 pthread join(tid[0], 0);
  pthread join(tid[1], 0);
```

main Stack i = 2Sema = 0thread1 Vargp "Thread 1" Sem wait Myid = 1thread2 Vargp "Thread 2" Myid = 2heap

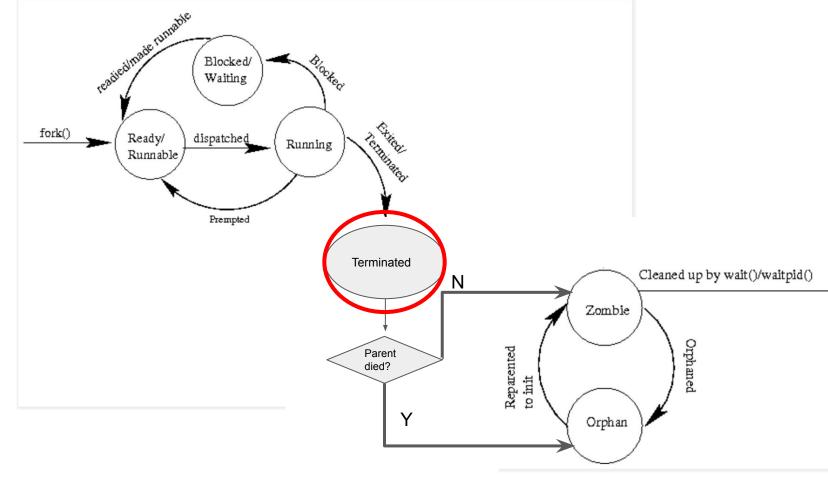
- Worksheet
- quizizz
- Fill up midterm review form
- Code review
- Coding trick & Style workshop

Q3

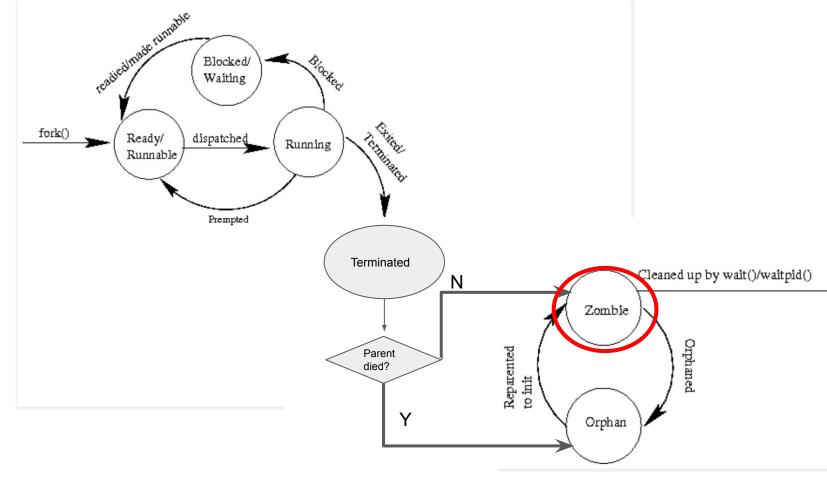


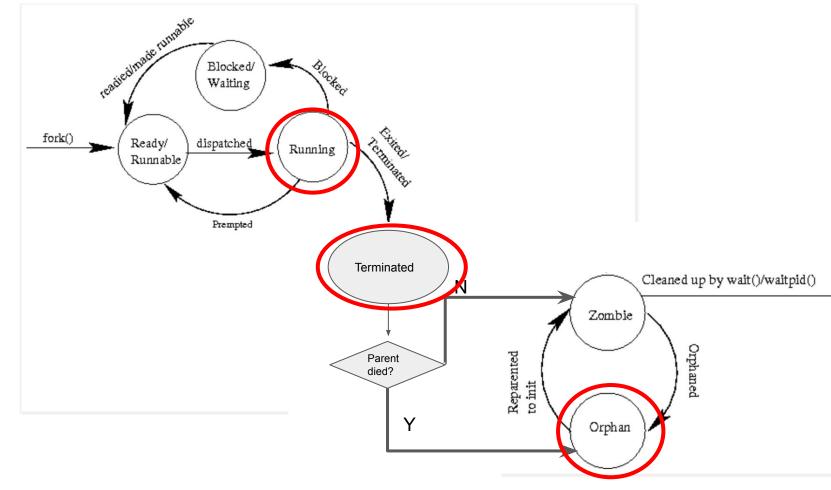
https://sites.ualberta.c a/dept/chemeng/AIX-43/share/man/info/C/ a\_doc\_lib/aixprggd/ge nprogc/fdescript.htm











- Worksheet
- quizizz
- Code review
- Coding Trick & Style workshop
- Fill up midterm review form

- Discuss your Best code and Worst Code with your neighbors
  - comments in code
  - writing modular code
  - good naming conventions
  - proper spacing and indentation, etc.

- Worksheet
- quizizz
- Code review
- Coding Trick & Style workshop
- Fill up midterm review form

- Worksheet
- quizizz
- Code review
- Coding trick & Style workshop
- Fill up midterm review form