Multiprocessing: threads (continued)

COSC 208, Introduction to Computer Systems, 2021-11-17

Announcements

- Exam 3
  - Take-home coding
- Project 3 due Thursday, December 2

Outline

- Warm-up
- pthreads API
- Race conditions

Warm-up

Q1: *What are all possible outputs produced by this program?*

```
void *thread1_main(void *arg) {
    int *x = (int *)arg;
    *x += 1;
    printf("x is %d\n", *x);
    return NULL;
}
void *thread2_main(void *arg) {
    int *y = (int *)arg;
    *y -= 1;
    printf("y is %d\n", *y);
    return NULL;
}
int main() {
    int *z = malloc;
    *z = 0;
    // Start thread running thread1_main(z)
    // Start thread running thread2_main(z)
    // Wait for threads to finish
    printf("z is %d\n", *z);
}
```

🛑 **STOP HERE** after completing the warm-up; if you have extra time, please **skip ahead** to the extra practice.

Pthreads API

Q2: *What are all possible outputs produced by this program?*

```c
#include <pthread.h>
void *printer(void *arg) {
    char *ch = (char*)arg;
    printf("I am %c\n", *ch);
    return NULL;
}
int main() {
    pthread_t thread1, thread2;
    char *ch1 = malloc(sizeof(char));
    *ch1 = 'X';
    char *ch2 = malloc(sizeof(char));
    *ch2 = 'Y';
    pthread_create(&thread1, NULL, &printer, ch1);
    pthread_create(&thread2, NULL, &printer, ch2);
    pthread_join(thread1, NULL);
    pthread_join(thread2, NULL);
}
```

Q3: *What are all possible outputs produced by this program?*

```c
#include <pthread.h>
void *printer(void *arg) {
    char *ch = (char*)arg;
    printf("I am %c\n", *ch);
    return NULL;
}
int main() {
    pthread_t thread1, thread2;
    char *ch = malloc(sizeof(char));
    *ch = 'A';
    pthread_create(&thread1, NULL, &printer, ch);
    *ch = 'B';
    pthread_create(&thread2, NULL, &printer, ch);
    pthread_join(thread1, NULL);
    pthread_join(thread2, NULL);
}
```

Q4: *What are all possible outputs produced by this program?*

```c
#include <pthread.h>
void *printer(void *arg) {
    char *ch = (char*)arg;
    printf("I am %c\n", *ch);
    return NULL;
}
int main() {
    pthread_t thread1, thread2;
    char *ch = malloc(sizeof(char));
    *ch = 'P';
    pthread_create(&thread1, NULL, &printer, ch);
    pthread_join(thread1, NULL);
    *ch = 'Q';
    pthread_create(&thread2, NULL, &printer, ch);
    pthread_join(thread2, NULL);
}
```

🛑 **STOP HERE** after completing the above questions; if you have extra time, please **skip ahead** to the extra practice.

Race conditions

```c
#include <pthread.h>
#include <stdio.h>
void *deposit(void *arg) {
    int *balance = (int *arg)
    int tmp = *balance
    tmp += 100;
    *balance = tmp;
    return NULL;
}
void *withdraw(void *arg) {
    int *balance = (int *arg)
    int tmp = *(int *)balance
    tmp -= 50;
    *balance = tmp;
    return NULL;
}
int main() {
    pthread_t thrA, thrB;
    int *balance = malloc(sizeof(int));
    *balance = 250;
    pthread_create(&thrA, NULL, &deposit, balance);
    pthread_create(&thrB, NULL, &withdraw, balance);
    pthread_join(thrB, NULL);
    pthread_join(thrA, NULL);
    printf("Balance: $%d\n", *balance);
}
```

Extra practice

Q5: *What are all possible outputs produced by this program?*

```c
#include <stdio.h>
#include <pthread.h>
void *printer2(void *arg) {
    char *ch = (char*)arg;
    printf("Start %c\n", *ch);
    printf("End %c\n", *ch);
    return NULL;
}
int main() {
    pthread_t thread1, thread2;
    char ch1='X', ch2='Y';
    pthread_create(&thread1, NULL, &printer2, &ch1);
    pthread_create(&thread2, NULL, &printer2, &ch2);
    pthread_join(thread1, NULL);
    pthread_join(thread2, NULL);
}
```