



Homework 3

Lectures 9, 10

Operating Systems

Dr. Javadi

Spring 2023



۱- با فرض اینکه pid های واقعی پردازش پدر و فرزند به ترتیب ۲۶۰۰ و ۲۶۰۳ باشد، خروجی خط های A,B,C,D برنامه زیر را با ذکر دلیل عنوان کنید.

```
#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>

int main()
{
    pid_t pid, pid1;

    /* fork a child process */

    if (pid < 0) { /* error occurred */
        fprintf(stderr, "Fork Failed");
    }
    else if (pid == 0) { /* child process */
        pid1 = getpid();
        printf("child: pid = %d", pid); /* A */
        printf("child: pid1 = %d", pid1); /* B */
    }
    else { /* parent process */
        pid1 = getpid();
        printf("parent: pid = %d", pid); /* C */
        printf("parent: pid1 = %d", pid1); /* D */
        wait(NULL);
    }

    return 0;
}
```



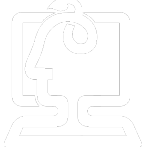
۲- فرض کنید کد زیر در یک ماشین لینوکس کامپایل و اجرا شده است. همچنین فرض کنید تمامی فراخوانی های سیستمی با موفقیت اجرا می شوند.

```
int main() {
    int count = 0;
    int pid=0, pid2=0;

    if ( (pid = fork()) ) {
        count = count + 2;
        printf("%d", count);
    }
    if (count == 0)
    {
        count++;
        pid2=fork();
        printf("%d", count);
    }

    if (pid2 || pid) {
        wait(NULL);
        count = count+4;
    }
    printf("%d", count);
}
```

درخت اجرای پردازش ها را برای این برنامه رسم کنید. همچنین مشخص کنید که هر پردازش هنگام رسیدن به توابع printf چه مقادیری را چاپ می کنند.



۳- کد C زیر را در نظر بگیرید، هنگامی که زمان اجرای خط B برسد، چه تعداد رشته (thread) در سیستم واسطه این برنامه فعال خواهد بود؟

```
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>

int value = 5;

void *print_message1(void *arg) {
    value += 2;

    char *message = "Hello from thread 1!"
    printf("%s\n", message); // LINE B
    pthread_exit(NULL);
}

void *print_message2(void *arg) {
    value += 2;

    char *message = "Hello from thread 2!"
    printf("%s\n", message); // LINE C
    pthread_exit(NULL);
}

int main() {
    pthread_t threads[2];

    pthread_create(&threads[0], NULL, print_message1, NULL);
    pthread_create(&threads[1], NULL, print_message2, NULL);

    pthread_join(threads[0], NULL);
    pthread_join(threads[1], NULL);

    printf("Both threads have completed. Value = %d\n", &value);

    return 0;
}
```



۴- کد C زیر را در نظر بگیرید:

```
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>

#define NUM_THREADS 2

int shared_value = 0;

void *increment_value(void *arg) {
    int *my_id = (int*) arg;
    int i;

    for (i = 0; i < 1000000; i++) {
        shared_value++;
    }

    printf("Thread %d has finished.\n", *my_id);
    pthread_exit(NULL);
}

int main() {
    int i, thread_ids[NUM_THREADS];
    pthread_t threads[NUM_THREADS];

    for (i = 0; i < NUM_THREADS; i++) {
        thread_ids[i] = i;
        pthread_create(&threads[i], NULL, increment_value, (void*) &thread_ids[i]);
    }

    for (i = 0; i < NUM_THREADS; i++) {
        pthread_join(threads[i], NULL);
    }

    printf("Final shared value: %d\n", shared_value);

    return 0;
}
```

الف) خروجی این کد را پیش بینی کنید و توضیح دهید که چرا فکر می کنید این خروجی را تولید می کند.

ب) در رابطه با کد داده شده، تفاوت بین استفاده از رشته ها و استفاده از فورک برای رسیدن به موازی سازی را توضیح دهید. به طور خاص، هنگام استفاده از فورک، آیا پردازش فرزند مقدار `shared_value` را برای هر دو فرآیند فرزند و والد تغییر می دهد؟ پاسخ خود را توضیح دهید. چرا هنگام استفاده از رشته ها متفاوت است؟



۵- فرض کنید یک پردازش دستور $value = value + 1$ را اجرا می کند. همچنین یک پردازش دیگر به صورت همروند و مستقل دستور $value = value - 2$ را اجرا می کند. اگر $value$ یک متغیر مشترک بین این دو پردازش باشد و فقط در این دو دستور استفاده شده باشد و مقدار این ابتدایی این متغیر $value = 5$ باشد، همه مقادیر ممکن برای متغیر $value$ بعد از اجرای این دو پردازش را بنویسید.



۱- برنامه ای بنویسید که از pthread برای محاسبه مجموع یک ماتریس $M \times N$ استفاده کند. برنامه باید M رشته ایجاد کند، جایی که هر رشته مجموع یک ردیف از ماتریس را محاسبه می کند. پس از اتمام اجرای همه رشته ها، رشته اصلی باید مجموع جزئی محاسبه شده توسط هر رشته را جمع کند تا مجموع نهایی ماتریس به دست آید.

پس از اجرای نسخه موازی، کد را تغییر دهید تا جمع بندی با استفاده از یک رشته (یعنی بدون استفاده از pthread) انجام شود. سپس، زمان اجرا دو پیاده سازی را مقایسه کنید و تفاوت هایی که مشاهده می کنید را توضیح دهید.

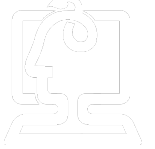
امتیازی

سوال فوق را برای ضرب ماتریس ها پیاده سازی کنید. توجه کنید که تعداد رشته ها می تواند هر عددی باشد و نیاز نیست حتما مقدار M را اختیار کند. از هر الگوریتم ضربی می توانید استفاده کنید به شرطی که توانایی موازی سازی عملیات وجود داشته باشد.

۲- به شما آرایه ای از اعداد صحیح داده می شود و باید برنامه ای را پیاده سازی کنید که آرایه را با استفاده از یک رشته مرتب می کند و با استفاده از رشته دیگر مقدار خاصی را در آرایه مرتب شده جستجو می کند. می خواهید بررسی کنید که آیا ترتیب اجرای رشته ها بر زمان اجرای برنامه تأثیر می گذارد یا خیر. یک کد C بنویسید که از دو رشته برای مرتب سازی و جستجوی یک مقدار در آرایه استفاده می کند. ممکن است فرض کنید که آرایه از قبل مقداردهی اولیه شده است.

سپس به سوالات زیر پاسخ دهید:

۱. اگر رشته مرتب سازی ابتدا اجرا شود، چگونه می تواند بر زمان اجرا تأثیر بگذارد در مقایسه با زمانی که رشته جستجو ابتدا اجرا شود؟
۲. اگر رشته جستجو ابتدا اجرا شود، آیا تابع جستجو می تواند روی یک آرایه مرتب نشده اجرا شود؟
۳. چه مراحل را می توانید انجام دهید تا مطمئن شوید که رشته مرتب سازی قبل از شروع رشته جستجو تمام می شود؟ چگونه ممکن است این روی زمان اجرای برنامه تأثیر بگذارد؟
۴. آیا می توانید راه دیگری برای بهینه سازی زمان اجرای برنامه در نظر بگیرید؟
۵. فرض کنید یک آرایه بزرگ با تکرارهای زیادی از مقدار جستجو دارید. چگونه ممکن است این روی زمان اجرای برنامه تأثیر بگذارد؟



به نکات زیر توجه کنید.

- مهلت ارسال تمرین ساعت ۲۳:۵۹ روز یکشنبه ۳ اردیبهشت ماه می باشد.
- در صورت کشف تقلب نمره تمرین ۰ در نظر گرفته می شود.
- سوالات خود را می توانید از طریق تلگرام از تدریس‌یارهای گروه خود بپرسید.
- فایل پاسخ تمرین را تنها با قالب **HW?_StudentNumber.pdf** در کورسز بارگزاری کنید.
- نمونه: HW2_9831072