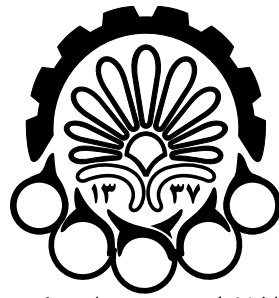


به نام خدا



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

سیستم‌های عامل (پاییز ۱۴۰۰)

تمرین دوم

استاد درس:

آقای دکتر جوادی

مهلت نهایی ارسال پاسخ:

۱۷ آبان ۱۴۰۰ ساعت ۲۳:۵۹

نکته مهم: دقت کنید که تمدید نخواهیم داشت و صرفاً می‌توانید ۱ تا ۵ روز از ۱۵ روز مجاز برای تاخیر ارسال تمامی تمرین‌های تئوری در این ترم را استفاده کنید. اگر بودجه ۱۵ روز شما تمام شود، به ازای هر روز تاخیر ۱۰ درصد از نمره تمرین را از دست خواهید داد.

لطفاً به سوالات زیر با دقت پاسخ دهید و پاسخ نهایی و کامل خود را در سایت درس بارگذاری کنید.

۱) استفاده از فراخوانی سیستمی `fork()` روشی پرکاربرد برای ایجاد پردازش فرزند در سیستم عامل‌های بر پایه POSIX است. به طور خلاصه پس از اجرای این دستور، پردازش فرزند که در آغاز کارش یک کپی از پردازش پدر است، به شکل همروند (یا همزمان) با پردازش پدر اجرا می‌شود. مهم‌ترین تفاوت در نحوه رفتار `fork()` است که در پردازش پدر مقدار شناسه یکتای پردازش فرزند (عدد بزرگتر از صفر) را برمی‌گرداند و در پردازش فرزند عدد صفر را برمی‌گرداند. همچنین اگر مقدار برگشت داده شده، عددی کمتر از صفر باشد به این معنی است که بر اثر رخداد خطایی، پردازش فرزند ایجاد نشده است. با این توصیفات اگر قطعه کد زیر اجرا شود، به ترتیب، خطوی که در پردازش پدر و فرزند چاپ می‌شوند، چه خواهد بود؟ دلیل پاسخ خود را به اختصار توضیح دهید.

```
int main() {
    pid_t = pid
    a = 30; /* address of a in memory is 2000 */
    pid = fork();
    if (pid < 0) {
        fprintf(stderr, "fork failed");
        return 1;
    } else if (pid == 0) {
        a = a + 10;
        printf("%d, %dn", a, &a);
    } else {
        a = a - 10;
        printf("%d, %dn", a, &a);
    }
    return 0;
}
```

۲) خروجی کد زیر چیست؟ نمودار (درخت) پردازش‌های آن را رسم کرده و درمورد آن به اختصار توضیح دهید.

```
#include <stdio.h>
#include <unistd.h>

int main()
{
    if ((fork() && !fork()) || (fork() && fork()))
        printf("3 ");
    return 0;
}
```

۳) در مورد نحوه بوجود آمدن پردازش‌های زامبی (Zombie) و یتیم (Orphan) توضیح دهید. نحوه رسیدگی سیستم عامل به آن‌ها به چه شکل است؟

۴) زمانبند پردازش به این نحو عمل می‌کند که هر پردازش ۱۵۰ کلاک برای اجرای روی CPU زمان دارد و بعد از پایان زمانش روی صف انتظار می‌رود که در لحظه ابتدایی (صفر) ۴ پردازش در صف موجود است و CPU خالی است. پردازش اول بعد از ۲۰۰ کلاک اجرای خود به فراخوانی سیستمی fork() می‌رسد. پردازش دوم بعد از هر ۱۰۰ کلاک به یک دستور i/o می‌رسد که ۴۰۰ کلاک زمان خواهد برد، پردازش سوم تنها به ۴۰۰ کلاک نیاز دارد تا به پایان برسد، پردازش چهارم بعد از ۵۰ کلاک نیاز به خواندن از حافظه اصلی دارد که ۵۰ کلاک زمان می‌برد. صف پردازش‌ها پس از گذشت ۱۰۰۰ کلاک به چه نحوی خواهد بود؟

Process4	Process3	Process2	Process1
----------	----------	----------	----------

CPU

۵) تکه کد زیر چند بار عبارت "hello" را چاپ می کند؟ (با این فرض که تمامی فراخوانی های سیستمی با موفقیت اجرا شوند)

```
...
{
    if(fork() && fork())
    {
        fork();
    }

    if(fork() || fork())
    {
        fork();
    }

    printf("hello");
    return 0;
}
```

موفق باشید

تیم درس سیستم های عامل