

ای نام تو بهترین سرآغاز
دستور کار سوم آزمایشگاه سیستم عامل
حمید فدیشه‌ای، دانشگاه بجنورد

آزمایش ۱-۳

برنامه‌ای به زبان C بنویسید که با استفاده از فراخوانی سیستمی `exec` یک فرایند جدید ایجاد نماید. تحقیق کنید که آیا واقعا فرایند جدید جایگزین فرایند اولیه می‌گردد؟ (مثلا دستور `printf` که بعد از `exec` موفق آمده باشد انتظار نداریم اجرا شود).
یادآوری:

فراخوانی سیستمی `exec` گونه‌های مختلفی دارد. یکی از آن‌ها `execl` است که به عنوان آرگومان مسیر فایل اجرایی فرایند را به همراه رشته پارامترهای خط فرمان مورد نظر اجرا دریافت می‌نماید. مثلا:
`execl("/usr/bin/ls", "/home", NULL);`
فرایند جدیدی از برنامه اجرایی `/usr/bin/ls` ایجاد می‌نماید.

آزمایش ۲-۳

برنامه‌ای به زبان C بنویسید که با استفاده از فراخوانی سیستمی `fork` یک فرایند جدید ایجاد نماید. تحقیق نمایید آیا واقعا فرایندها به طور موازی اجرا می‌شوند؟ (مثلا خروجی `printf` های مکرر به فواصل زمانی خاص در فرزند و والد باید همزمان و در نتیجه مخلوط شده با هم دیده شوند).
یادآوری:

با انجام فراخوانی سیستمی `fork` یک فرایند فرزند ایجاد می‌شود و والد و فرزند هر دو ادامه می‌یابند. همیشه مقدار برگشتی `fork` در صورت خطا -1 است. در صورت موفقیت در فرایند فرزند مقدار صفر و در فرایند والد مقدار شناسه فرایند فرزند باز می‌گردد. مثلا

```
pid_t pid = fork();
if (pid == -1)
{
    // print error
}
else if (pid == 0)
{
    // put child code here
}
else
{
    // continue parent here
}
```

آزمایش ۳-۳

در آزمایش fork وجود یا عدم وجود اشتراک داده بین فرایندها را بررسی نمایید. (یک آرایه سراسری تعریف کنید. با یک آزمایش بررسی کنید آیا اگر فرایند والد تغییری در محتوای آرایه ایجاد کند فرایند فرزند تغییر را مشاهده می‌کند؟)

آزمایش ۴-۳

برنامه ای بنویسید که یک فرایند دو ریسمان اجرا ایجاد نماید. برای ایجاد ریسمان جدید از فراخوانی pthread_create استفاده نمایید. در این فراخوانی نام تابعی که مایلید یک ریسمان اجرایی از آن ساخته شود به pthread_create داده می‌شود. مثلا اگر تابع زیر را نوشته باشید:

```
void *thread1_func(void *arg)
{
    // code of thread 1
}
```

برای ایجاد یک ریسمان جدید از اجرای این تابع باید:

```
pthread_t thread1;
pthread_create(&thread1, NULL, thread1_func, NULL);
```

آزمایش ۵-۳

در مورد آزمایش ایجاد ریسمان بررسی کنید که آیا اشتراک داده بین دو ریسمان اجرا وجود دارد؟

تمرین ۳

یک ماتریس در قالب یک آرایه با ۵ سطر و ۶ ستون وجود دارد. یک برنامه چندریسمانی برای محاسبه مجموع اعضای این آرایه بنویسید. هر ریسمان وظیفه جمع زدن عناصر یکی از سطرهای ماتریس را بر عهده دارد. در نهایت برنامه اصلی حاصل محاسبه شده توسط همه ریسمان‌ها را با هم جمع زده نتیجه را در خروجی نمایش دهد. نکته: در صورت نیاز به انتظار تا اتمام اجرای ریسمانی خاص، می‌توانید از فراخوانی pthread_join استفاده نمایید. مثلا فراخوانی زیر

```
pthread_join(thread1, NULL);
```

فراخواننده را مسدود می‌نماید تا زمانی که اجرای ریسمان thread1 پایان یابد.