



سیستم عامل

تمرین دوم عملی

طراح: حامد فیض آبادی

آبان ۱۴۰۱

فهرست مطالب

۲	۰ نکات
۳	۱ سورت آرایه
۳	۱.۱ سورت آرایه با process
۳	۲.۱ سورت آرایه با thread
۳	۲ web server
۳	۱.۲ fork approach
۴	۲.۲ thread approach
۴	۳.۲ مقایسه روش ها
۴	۳ خواندن محتویات فایل
۴	۱.۳ fork+exec
۴	۲.۳ system

نکات

- کدهای ارسالی شما باید به زبان c باشد و cpp و زبانهای دیگر قابل قبول نیست. همچنین تنها مجاز به استفاده از کتابخانههای زبان c که برای لینوکس است، هستید.
- در این تمرین باید برای هر سوال یک makefile بنویسید و در گزارشی که همراه تمرین ارسال میکنید به صورت خلاصه بگویید کد شما باید چگونه اجرا شود و چه ورودیهایی به آن میدهیم و چه خروجیهایی دارد. در گزارش نحوه ران گرفتن از کد باید توضیح داده شود یا اگر سوال توضیحی پرسیده شده بود، پاسخ داده شود و نیازی به توضیح منطق پیادهسازی کدها نیست.
- در صورت وجود شباهت غیرعادی بین پاسخهای ارسالی مطابق با قوانین درس نمره تمرین محاسبه خواهد شد.
- کد هر سوال را در فولدر مربوط به خودش قرار دهید و در نهایت تمام فولدرها همراه با گزارش را زیپ کرده و با نام HW2_student-id.zip ارسال کنید.

۱ سورت آرایه

۱.۱ سورت آرایه با process

یک آرایه با سایز n را به m قسمت مساوی تقسیم کنید (n بر m بخش پذیر است) و با استفاده از سیستم کال `vfork` (با `fork` فرق میکند) قسمت‌های آن را جداگانه در هر فرایند مرتب کنید و در نهایت با مرج سورت به صورت کلی آرایه را مرتب کنید. در انتخاب مقادیر n ، m و الگوریتم مرتب سازی هر زیر بخش از آرایه آزادی عمل دارید فقط باید کد شما طوری باشد که بتوان مقادیر n ، m را تغییر داد و همچنان کد کار کند، به همین منظور سعی کنید این متغیرها را در ابتدای برنامه تعریف کنید و به آن‌ها در کد ارجاع بدهید.

۲.۱ سورت آرایه با thread

بخش اول را با استفاده از `thread` ها پیاده سازی کنید. در واقع شما یک `process` در اینجا با چند `thread` دارید که هر کدام یک زیربخش از آرایه را با الگوریتم انتخابی شما مرتب میکند و در نهایت `thread` اصلی با مرج سورت این زیربخش‌ها را ادغام میکند.

۲ web server

وب سرور برنامه‌ای است که می‌تواند درخواست‌ها را از طریق وب مدیریت کند. یکی از ویژگی‌هایی که یک وب سرور باید داشته باشد این است که می‌تواند به بسیاری از افراد در لحظه با تاخیر کم پاسخ دهد. به عنوان مثال گوگل را به عنوان وب سروری در نظر بگیرید که میلیون‌ها کلاینت دارد و می‌تواند با تاخیر خیلی کمی به همه‌ی آن‌ها و به صورت همزمان پاسخ دهد. به این تمرین یک برنامه وب سرور بسیار ساده پیوست شده است. این وب سرور تنها قابلیت این را دارد که یکی یکی به کلاینت‌ها پاسخ بدهد. کد این وب سرور را طوری تغییر بدهید که بتوانیم به صورت موازی به کلاینت‌ها پاسخ بدهیم.

۱.۲ fork approach

- Step 1: Compile and Run the server.c
- Step 2: Type in your browser: `http://localhost:8090/`
- Step 3: Add `fork` to `server.c`; Then goto step 1.
- If you get the error: "In bind: Address already in use" try to change PORT number in the file and then recompile and run.

۲.۲ thread approach

- Step 1: Compile and Run the server.c
- Step 2: Type in your browser: `http://localhost:8090/`
- Step 3: Include pthread library and add necessary changes to server.c; Then goto step 1.
- If you get the error: "In bind: Address already in use" try to change PORT number in the file and then recompile and run.

۳.۲ مقایسه روش‌ها

بعد از پیاده سازی دو روش بالا باید کمی بیشتر با مفهوم multi-processing و multi-threading آشنا شده باشید. در گزارش خود این دو روش را با هم مقایسه کنید و مزیت‌ها و معایبشان را بگویید. همچنین بگویید شکل memory در هر کدام از روش‌ها چگونه است.

۳ خواندن محتویات فایل

یک فایل txt درست کنید و در آن چند خط بنویسید. برنامه cat در لینوکس میتواند محتویات فایل را نمایش دهد. ابتدا بدون نوشتن کدی سعی کنید با کامند `cat file_name.txt` محتویات فایل را در ترمینال مشاهده کنید.

۱.۳ fork+exec

یک برنامه بنویسید که به عنوان ارگيومنت نام یک فایل متنی را میگیرد و با استفاده از سیستم کال fork یک فرآیند فرزند می‌سازد و با خانواده exec برنامه cat را در فرآیند فرزند اجرا میکند و محتویات فایل متنی را به عنوان خروجی میدهد. فرآیند پدر باید صبر کند که فرآیند فرزند تمام شود. کد شما باید به این صورت قابل اجرا باشد:

```
./read_file.o file_name.txt
```

۲.۳ system

درباره‌ی سیستم کالی به نام system تحقیق کنید و نحوه کار آن را در گزارش توضیح دهید و سپس سعی کنید کد بخش بالا را با استفاده از آن بنویسید.