



دانشکده مهندسی کامپیوتر

درس سیستم‌های عامل

تمرین سری دوم

مدرسین دکتر رضا انتظاری ملکی

تیم طراحی رضا حقیقت گو، کامیار مرادیان زه آب

تاریخ انتشار ۱۴۰۳/۰۱/۲۲

تاریخ تحویل ۱۴۰۳/۰۲/۰۴

در رابطه با تمرین

➤ این تمرین شامل مباحث:

• Thread

• Process

می باشد.

➤ نمره این تمرین از 100 می باشد و بارم هر سوال روبه روی آن نوشته شده است.

➤ به هیچ وجه تمرینی را از دیگران کپی نکنید. در صورت مشاهده تقلب و کپی در تمرینات، نمره هر دو طرف صفر در نظر گرفته می شود.

۱. سورت بیشینه‌ها

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) m فرآیند را ایجاد کنید. در هر فرآیند بیشینه مقدار را برای بخش مربوطه بدست آورید. سپس در فرآیند اصلی این مقادیر بیشینه را به صورت نزولی مرتب کنید. در انتخاب مقادیر m, n و الگوریتم مرتب سازی هر زیر بخش از آرایه آزادی عمل دارید، فقط باید کد شما طوری باشد که بتوان مقادیر m و n 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 در اینجا یک `process` با چند `thread` دارید که هر کدام مقدار بیشینه زیربخش مربوط به خود را با الگوریتم انتخابی شما مرتب می‌کند و در نهایت `thread` اصلی با مرج سورت این مقادیر بیشینه بدست آمده را به صورت نزولی سورت می‌کند.

۲. جست و جوی فایل

در این سوال قصد داریم یک برنامه برای `search` در قسمت فایل های سیستمی بنویسیم. برنامه در ابتدا نام یک فایل را از کاربر دریافت می‌کند و در نهایت تمام مسیر هایی که فایلی با این نام در سیستم شما وجود دارد را باز می‌گرداند. برنامه شما باید قابلیت این را دارا باشد که از `root` سیستم شروع به جست و جو کند و در صورتی که فایلی مشابه با نام ورودی کاربر در پوشه مربوطه موجود بود، مسیر آن را نمایش دهد و اگر که پوشه ی دیگری در آن قرار داشت به روش **multi-thread** آن پوشه را نیز جستجو کند و در پوشه های بعدی هم همین مراحل تکرار می‌شوند.

توجه داشته باشید که این سوال تنها باید با روش `multi-thread` و به سایر راه حل ها نمره ای تعلق نخواهد گرفت. همچنین تنها مجاز به استفاده از کتابخانه هایی هستید که صرفا محتویات پوشه را نمایش دهد و الگوریتم سرچ را خودتان پیاده سازی کنید. نتیجه باید به این صورت قابل مشاهده باشد:

```
./search / fie_name
```

۳. مدیریت فایل‌ها

یک فایل txt درست کنید و در آن به طور متوالی در چند خط، عدد بنویسید به طوری که در هر خط تنها یک عدد قرار داشته باشد. تعداد اعداد موجود در این فایل حداقل ۱۰۰ باشد. حال نیاز است که یک برنامه جهت محاسبه میانگین این اعداد بنویسید. در این برنامه از Multi-Threading جهت تسریع عملکرد برنامه استفاده می‌کنیم. پس از محاسبه میانگین اعداد با استفاده از سیستم کال fork یک فرآیند فرزند بسازید و با استفاده از خانواده exec و با استفاده از mkdir یک فولدر با نام مشخص شده در آرگومان این برنامه بسازید. سپس در داخل این فولدر، با استفاده از دستور touch، یک فایل با نامی معادل با شماره دانشجویی خود بسازید و میانگین محاسبه شده را در آن بنویسید. در تمام طول اجرای برنامه، فرایند پدر باید صبر کند که فرایند فرزند تمام شود. کد شما باید به این صورت قابل اجرا باشد:

```
./program.out dir_name
```

۴. Download Manager

یکی از مشکلات دانلود چند فایل به صورت همزمان، ایجاد صف است که سبب می‌شود در صورتی که مشکلی در ارسال پکت‌های فایل جلوی صف پیش بیاید فایل‌های دیگر زمان زیادی را در صف طی کنند در حالی که اگر آنها در صف جلوتر بودند می‌توانستند بدون مشکل پکت‌ها را دریافت کنند.

جهت رفع این مشکل، شما برنامه‌ای بنویسید که آدرس یک فایل که لینک‌های دانلود در آن قرار دارد و همچنین محل دانلود فایل‌ها را به عنوان ورودی بگیرد و شروع به دانلود فایل‌ها به صورت multi-thread کند. برای دانلود فایل‌ها در C، پیشنهاد می‌شود از کتابخانه <curl/curl.h> استفاده کنید. برنامه شما باید به این صورت قابل اجرا کردن باشد:

```
./download_manage path/to/download/links path/to/download/directory
```

۵. درخت جست‌وجو

در این سوال قصد داریم تفاوت میان جست و جوی عادی یک گراف و جست‌وجوی multi-process آن را بررسی کنیم. جهت این امر ابتدا یک struct ساده از یک node را ایجاد کنید. برای مثال می‌توانید از این کد استفاده کنید:

```
struct Node {  
    int id;  
    int value;  
    struct Node* children;  
}
```

سپس به صورت دستی به آن مقدار بدهید و سعی کنید یک درخت را به این روش ایجاد کنید. درخت دلخواه شما باید حداقل از Node 10 تشکیل شده باشد و node ها ترجیحا به صورت متوازن پخش شده باشند. حال ابتدا یک بار بدون استفاده از multi-process به روش DFS، یک مقدار مانند id یکی از برگ های درخت را جست و جو کنید. سپس همین عمل را این بار از روش multi-process انجام دهید. زمان هریک از دو جست و جو را در ترمینال نمایش دهید. سپس پاسخ سوالات زیر را در کد خود کامنت کنید.

- زمان هر دو روش را باهم مقایسه کنید و نتیجه را ارزیابی کنید.
- بنظر شما اگر که تعداد node ها خیلی بیشتر بود، نتیجه به چه صورت می شد؟

۶. وب سرور آنالیز اعداد

وب سرور برنامه ای است که می تواند درخواست ها را از طریق وب مدیریت کند. یکی از ویژگی هایی که در یک سرور باید وجود داشته باشد، آن است که بتواند در مدت زمان کوتاهی به تعداد زیادی از افراد با تاخیر خیلی کم پاسخ دهد. به عنوان مثال گوگل را به عنوان وب سروری در نظر بگیرید که میلیون ها کلاینت دارد و می تواند با تاخیر خیلی کمی به همه آنها و به صورت همزمان پاسخ دهد. هدف از این سوال این است که یک وب سرور برای آنالیز عددی را ایجاد کنید به طوری که بتواند به طور موازی به کاربران پاسخ دهد. وظیفه این وب سرور آن است که یک مجموعه از اعداد را از سمت کلاینت دریافت کند، و برای هر مجموعه [انحراف معیار](#) کل مجموعه را بدست آورد. برای آنکه سرور بتواند به صورت موازی به درخواست ها پاسخ دهد، از Multi-Threading استفاده می شود. همچنین برای تسریع در روند محاسبات از Multi-Processing استفاده خواهد شد. در ذیل روند کلی درخواست و پاسخ این وب سرور آورده شده است.

- سمت سرور یک فرآید پیوسته در حال اجرا باشد و منتظر دریافت درخواست از سمت کلاینت باشد.
- پس از آنکه درخواست از سمت سرور پذیرفته شد، یک Thread جدید برای پاسخگویی به درخواست کلاینت به وجود خواهد آمد.
- با ایجاد Thread جدید و برقراری ارتباط میان سرور و کلاینت به طور موفقیت آمیز، نیاز است که از سمت کلاینت تعدادی عدد اعشاری ارسال شوند.

○ پیشنهاد تعداد اعداد را ۱۰۰ در نظر بگیرید.

- برای ارسال اعداد از نوع داده ای String استفاده کنید به طوری که بین هر عدد یک فاصله قرار گرفته باشد. به عنوان مثال مجموعه اعداد زیر را در نظر بگیرید:

"1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0"

- پس از دریافت این اعداد توسط سرور، محاسبات لازم جهت محاسبه انحراف معیار انجام خواهد شد.
- در ابتدا میانگین کل اعداد Thread محاسبه می شود.
- سپس با توجه به طول مجموعه ورودی، تعدادی فرآیند ایجاد خواهند شد. علاوه بر این نیاز است که یک حافظه مشترک برای تمام فرآیندها ایجاد شود. وظیفه هر فرآیند آن است که بخش صورت فرمول انحراف معیار را به ازای هر عدد موجود در مجموعه محاسبه کند و در حافظه مشترک قرار دهد.
- در پایان در خود Thread تمام مقادیر بدست آمده برای صورت فرمول را از داخل حافظه مشترک دریافت کرده و با هم جمع می کند و انحراف معیار برای مجموعه اعداد محاسبه خواهد شد و برای کلاینت ارسال می شود.
- برای آنکه بتوانید چندین عدد را از ورودی دریافت کنید، می توانید از تابع strtok استفاده کنید. برای بررسی اینکه کاربران می توانند به صورت موازی به سرور متصل شوند، می توانید از تابع sleep استفاده کنید تا میان پاسخ دادن به کاربران فاصله بیوفتد.

همچنین یک فایل برای بخش سرور به فایل‌های مربوط به این تمرین پیوست شده است که نیاز به تکمیل شدن جهت عملی شدن موارد گفته شده دارد. برای نوشتن بخش کلاینت نیز می‌توانید از کد مربوط به بخش سرور ایده بگیرید و آن را پیاده‌سازی کنید. در پایان هر دو فایل مربوط به بخش‌های کلاینت و سرور را آپلود کنید.