

تمرین سری سوم درس سیستم‌های عامل

فصل ۴ کتاب

موعد تحویل: ۱۳۹۴/۰۷/۲۷

- این تمرین را به صورت گروه‌های دونفره انجام دهید.
- در این تمرین پردازش متفاوتی برای هر گروه در نظر گرفته شده است و لازم است یکی از اعضای گروه پردازش مورد نظر را در سایت درس انتخاب کند.
- کدها و توضیحات این تمرین را در سایت درس آپلود کنید.
- در صورت مشاهده‌ی هر گونه شباهت بین کدها نمره‌ی هر دو گروه ۰ در نظر گرفته می‌شود.

در این تمرین، تسریع‌سازی برخی محاسبات از طریق توزیع آن‌ها روی تعدادی ریسمان (thread) بررسی می‌شود. بدین منظور، می‌بایست تعدادی ریسمان ساخته شود و سپس قسمت‌هایی از محاسبات را که از یکدیگر مستقل هستند، به آن‌ها سپرده شود، در نهایت نتایج این محاسبات را جمع کرده و پاسخ نهایی را تولید نمایید. برای هر پردازش، حالت موازی و سری را مقایسه کرده، میزان تسریع گزارش گردد و نیز مقدار بهینه برای تعداد ریسمان‌ها را بدست آورید.

پرسش اول. آیا همیشه زیاد کردن تعداد ریسمان‌ها باعث افزایش سرعت برنامه می‌شود؟ با دلیل توضیح دهید.

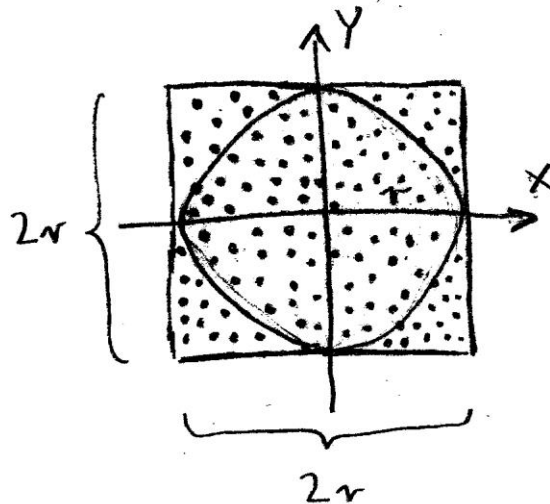
پرسش دوم. استفاده از ThreadPools چه مزایایی دارد؟ توضیح دهید.

توجه: پیاده سازی کد با استفاده از OpenMP نمره امتیازی خواهد داشت.

دو پردازش باید بررسی شود که یکی اجباری و دیگری به انتخاب خودتان خواهد بود:

الف) پردازش اجباری: محاسبه عدد پی (π) با استفاده از روش مونت کارلو

ایده اصلی این است که مساحت یک ناحیه با استفاده از شمردن تعداد نقطه‌هایی که بصورت تصادفی در آن ناحیه پخش شده‌اند، تقریب بزنیم. تعداد نقاطی که در داخل دایره واقع شده‌اند را با تعداد کل نقاط در کل ناحیه مقایسه خواهیم کرد.



با استفاده از این نواحی و دانستن فرمول مساحت دایره و مربع خواهیم داشت:

$$\frac{A_{circle}}{A_{square}} = \frac{\pi r^2}{(2r)^2} = \frac{\pi r^2}{4r^2} = \frac{\pi}{4}$$

بنابراین:

$$\pi = 4 * \frac{A_{circle}}{A_{square}}$$

ب) پردازش‌های انتخابی: یکی از پردازش‌های زیر را انتخاب نمایید. (انتخاب در گروه‌ها باید متمایز باشد)

۱. ضرب ماتریس‌ها

برنامه در این پردازش دو ماتریس به عنوان ورودی دریافت کرده و محاسبه ی هر درایه به یک ریسمان سپرده می شود. در نهایت ماتریس نهایی به عنوان خروجی چاپ می شود.

۲. مرتب سازی ادغامی آرایه

برنامه در این پردازش با دریافت یک آرایه آن را به دو قسمت مساوی تقسیم کرده و مرتب سازی هر قسمت را به یک

ریسمان می‌سپارد (توجه کنید هر ریسمان می‌تواند از الگوریتم مرتب سازی مستقلی استفاده کند.) در نهایت این ۲ قسمت مرتب شده را با یکدیگر ادغام کرده و خروجی را چاپ می‌کند. الگوریتم Merge Sort نمونه‌ای از این پردازش است.

۳. مرتب سازی سریع آرایه

در این پردازش، الگوریتم QuickSort انجام می‌شود که در آن تقسیم کردن پردازش به زیر پردازشها باید صورت گیرد. آرایه از ورودی دریافت می‌شود و هر بخش که بتواند بصورت همروند اجرا شود باید به ریسمان شکسته شود.

۴. BFS توزیع شده

برنامه در این پردازش با دریافت یک درخت با قالبی که ذکر خواهد شد و یک راس مبدا، الگوریتم BFS را روی زیر درخت‌های راس مبدا روی ریسمان‌های مختلف اجرا کند. این برنامه در نهایت حداقل فاصله‌ی راس‌های این درخت از راس مبدا را به عنوان خروجی باز می‌گرداند.
به طور مثال:

۳ → تعداد رئوس گراف

۱ ۲ ۳ → یال بین دو راس ۱ و ۲ با وزن ۳

۲ → راس مبدا

۵. فیبوناچی

برنامه در این پردازش با دریافت عدد n ، دست به محاسبه‌ی جمله‌ی n -ام فیبوناچی به این صورت می‌زند که جمله‌ی $n-1$ -ام و $n-2$ -ام را هر کدام را در یک ریسمان پردازش می‌کند (توجه کنید هر ریسمان می‌تواند از الگوریتم محاسبه‌ی مستقلی برای محاسبه جمله مورد نظر فیبوناچی استفاده کند). در نهایت با جمع ۲ عدد برگردانده شده جمله‌ی n -ام فیبوناچی تولید می‌شود. در ادامه در مورد کارایی (Performance) این کار در مقایسه با روش بدون ریسمان بحث کنید.

۶. یافتن بزرگترین عنصر یک آرایه

می‌خواهیم از بین n عدد که n توانی از ۲ می‌باشد، بزرگترین عدد را از طریق یک الگوریتم چند ریسمانی پیدا کنیم. الگوریتم به شرح زیر:

در این روش، اعداد در گروه‌های دوتایی جفت می‌شوند و بزرگترین عدد از هر جفت (برنده) یافته می‌شود. سپس برنده‌ها با هم جفت می‌شوند از هر جفت، برنده پیدا می‌شود و این روند همچنان ادامه می‌یابد تا یک کلید باقی بماند. با استفاده از پردازنده‌های بیشتر سود بیشتری می‌توانید از این روش ببرید. تعداد ریسمان‌های لازم فقط نصف تعداد عناصر آرایه است. هر ریسمان دو عنصر از آرایه را به درون متغیرهای first و second از حافظه محلی می‌ریزد. سپس عنصر بزرگتر را در محل اولی از آرایه‌ای می‌نویسد که عناصر را از آن خوانده است و عنصر دوم را دست نخورده باقی می‌گذارد.

۷. جمع ماتریس ها

برنامه در این پردازش دو ماتریس نسبتاً بزرگ را دریافت کرده و ماتریس حاصل از جمع آن ها را با توزیع محاسبات روی

تعدادی ریسمان محاسبه و چاپ می کند.

۸. خواندن فایل

یک فایل که شامل یک عدد دو رقمی در هر سطر است را در سیستم در اختیار است. هر ریسمان، این فایل را خوانده و باید عددی که در هر سطر مجزای این فایل قرار دارد را دریافت کند (هیچ دو ریسمانی نباید یک خط مشترک را بخوانند) و مقدار موجود در آن سطر را به شاخه اصلی برگرداند. در شاخه اصلی مقادیر بازگردانده شده جمع شده و در نهایت آن مقدار را چاپ کند.

۹. محاسبه حلقه‌های تو در تو

در این برنامه یک متغیر به عنوان ورودی در تعدادی حلقه تو در تو for داده می‌شود و محاسبه هریک از حلقه‌ها به یک ریسمان اختصاص می‌یابد. در نهایت مقدار متغیر چاپ می‌شود.
به عنوان مثال:

```
int numb=0;

for(i = 0; i < n; i++){
    for(j = 0; j < m; j++){
        for(k = 0; k < l; k++){
            for(x = 0; x < o; x++){
                numb++;
            }
        }
    }
}

printf("%d\n", numb);
```

۱۰. تعیین ساعت کشورها

در این برنامه در ورودی نام تعدادی کشور داده می‌شود. هر ریسمان باید با توجه به یک کشور که در ورودی دریافت شده است، ساعت سیستم را دریافت کند و ساعت کشور مورد نظر را باز گرداند.
به عنوان مثال:

(System time: 08:00)

Input:

Russia, USA, England, France, China, Japan

Output:

thread1: Russia: 07:30

thread2: USA: 00:30

etc

۱۱. تعیین نقاطی که در فاصله‌ی معینی از نقطه‌ی هدف قرار دارند

برنامه در این پردازش تعدادی نقطه در صفحه‌ی مختصات دو بعدی و یک نقطه‌ی هدف و فاصله‌ی d را دریافت کرده و

نقاطی که در فاصله‌ی d از نقطه‌ی هدف قرار دارند، به عنوان خروجی چاپ کند. با توجه به تعداد نقاط، صفحه‌ی مختصات به تعدادی بخش تقسیم می‌شود و بررسی نقاط هر بخش به یک ریسمان سپرده می‌شود.

۱۲. تعیین مرکز جرم

برنامه در این پردازش مشخصات تعدادی ذره در صفحه‌ی مختصات دو بعدی (جرم، طول و عرض) را به عنوان ورودی دریافت کرده و مختصات مرکز جرم آن‌ها را محاسبه و چاپ می‌کند. با توجه به تعداد نقاط، صفحه‌ی مختصات به تعدادی بخش تقسیم می‌شود و تعیین مرکز جرم هر بخش به یک ریسمان سپرده می‌شود و در نهایت مرکز جرم همه‌ی نقاط محاسبه می‌شود.

۱۳. محاسبه‌ی مجموع اعداد آرایه یک بعدی

برنامه در این پردازش یک آرایه‌ی یک بعدی بسیار بزرگ را دریافت کرده و مجموع اعداد آرایه را محاسبه و چاپ می‌کند. با توجه به طول آرایه، آرایه به تعدادی بخش تقسیم می‌شود و محاسبه‌ی مجموع هر بخش به یک ریسمان سپرده می‌شود.

۱۴. پیدا کردن کلمه مورد نظر

تعدادی فایل با محتوای متفاوت در اختیار داریم. در ورودی این برنامه یک کلمه را دریافت می‌کنیم. پس از آن، این کلمه را به ریسمان‌ها داده تا هر ریسمان یک فایل را بررسی کند که آیا کلمه مورد نظر در آن فایل موجود هست یا خیر، و اگر موجود هست تعداد دفعاتی که تکرار شده را باز گرداند.

۱۵. تعیین زمانی که از شروع سال گذشته

در این برنامه می‌خواهیم مدت زمانی که از شروع سال گذشته است را بررسی نماییم. به این صورت که هر ریسمان مدتی که از شروع سال گذشته است را با معیاری جداگانه برگرداند. به عنوان مثال یک ریسمان، این مدت را براساس ثانیه برگرداند و ریسمان دیگری بر اساس ساعت. این معیارها عبارت‌اند از: ماه، هفته، روز، ساعت، دقیقه و ثانیه.

۱۶. صاف کردن (smoothing) یک تصویر

برنامه در این پردازش یک تصویر را به عنوان ورودی دریافت کرده و smooth شده‌ی آن را به عنوان خروجی چاپ می‌کند (این روش در پردازش تصویر برای از بین بردن نویزها به کار می‌رود).

این کار با استفاده از محاسبه‌ی میانگین همه‌ی ۸ خانه اطراف پیکسل وسط و قرار دادن مقدار پیکسل وسط با مقدار به دست آمده صورت می‌گیرد. با توجه به اندازه‌ی تصویر، تصویر به تعدادی بخش تقسیم می‌شود و تبدیل هر بخش از تصویر به یک ریسمان سپرده می‌شود.

۱۷. محاسبه‌ی تعداد دفعات تکرار هر یک از اعداد در ماتریس

برنامه در این پردازش یک ماتریس دو بعدی از اعداد تصادفی بین ۱ تا ۱۰ تا دریافت کرده و تعداد دفعات تکرار هر یک از اعداد را در ماتریس محاسبه و چاپ می‌کند. با توجه به اندازه‌ی ماتریس، ماتریس به تعدادی بخش تقسیم می‌شود و محاسبه‌ی تعداد دفعات تکرار هر یک از اعداد در هر بخش به یک ریسمان سپرده می‌شود.

۱۸. یافتن کلمات با حروف غیر تکراری در یک فایل

برنامه در این پردازش یک فایل حاوی تعداد زیادی کلمه را دریافت کرده و کلماتی را که حروف آن‌ها غیرتکراری است را در یک فایل می‌نویسد. با توجه به تعداد کلمات، فایل به تعدادی بخش تقسیم می‌شود و پردازش هر بخش به یک ریسمان سپرده می‌شود.

۱۹. یافتن کلمات palindrome در یک فایل

برنامه در این پردازش یک فایل حاوی تعداد زیادی کلمه را دریافت کرده و کلماتی که palindrome هستند را در یک فایل می‌نویسد. با توجه به تعداد کلمات، فایل به تعدادی بخش تقسیم می‌شود و پردازش هر بخش به یک ریسمان سپرده می‌شود.

۲۰. یافتن اعداد اول در یک فایل

برنامه در این پردازش یک فایل حاوی تعداد زیادی عدد طبیعی را دریافت کرده و اعدادی را که اول هستند را در یک فایل می‌نویسد. با توجه به تعداد اعداد، فایل به تعدادی بخش تقسیم می‌شود و پردازش هر بخش به یک ریسمان سپرده می‌شود.

۲۱. یافتن تعداد دفعات تکرار حروف صدادار در یک فایل

برنامه در این پردازش یک فایل حاوی تعداد زیادی کلمه را دریافت کرده و تعداد دفعات تکرار هر یک از حروف صدادار در فایل را محاسبه و چاپ می‌کند. با توجه به تعداد کلمات، فایل به تعدادی بخش تقسیم می‌شود و پردازش هر بخش به یک ریسمان سپرده می‌شود.

۲۲. کپی کردن فایل

برنامه ای بنویسید که کپی کردن یک فایل را به صورت چند ریسمانی انجام دهد، هر ریسمان باید یک قسمت از فایل را در مقصد کپی کند. توجه کنید که قبل از کپی کردن فایل، باید فایل مقصد را ایجاد نمایید.

۲۳. کوچک کردن اندازه عکس

هر عکس یک ماتریس $n*m$ است که در هر عنصر شدت روشنایی آن نقطه از عکس مشخص شده است. برای کوچک کردن عکس با ضریب $2/1$ باید هر چهار خانه مجاور که در یک مربع قرار میگیرند را میانگین گرفته و در خانه متناظر آن در ماتریس مقصد قرار دهیم. برنامه‌ای بنویسید که ماتریس مقصد را به صورت چند ریسمانی از روی مبدا بسازد.

۲۴. دانلود کردن تعدادی فایل

برنامه‌ای بنویسید که با گرفتن لینک تعدادی فایل (۱۰۰۰۰۰۰ صد هزار عدد) به صورت چند ریسمانی تلاش برای دانلود کردن آنها کند. بهترین تعداد ریسمان برای انجام اینکار چقدر است؟

۲۵. یافتن کلماتی در یک فایل که طول آن‌ها از حد تعیین شده بیشتر است

برنامه در این پردازش یک فایل حاوی تعداد زیادی کلمه و C را دریافت کرده و کلماتی که طول آنها از C بیشتر است را در یک فایل می‌نویسد. با توجه به تعداد کلمات، فایل به تعدادی بخش تقسیم می‌شود و پردازش هر بخش به یک ریسمان سپرده می‌شود.

۲۶. ساختن ماتریس لبه‌های یک عکس

لبه در یک عکس به تغییرات شدید شدت روشنایی عکس گفته می‌شود. برای اینکار باید اختلاف عناصر مجاور در ماتریس عکس را محاسبه کرده و چنانچه این اختلاف از C بیشتر بود، پیکسل متناظر با آن را در ماتریس خروجی فعال کنید. برنامه‌ای بنویسید که C و عکس ورودی را گرفته و ماتریس خروجی را به صورت چند ریسمانی از روی ورودی بسازد.

۲۷. مسئله پخش گرما

فرض کنیم وقتی یک نقطه از یک صفحه را گرم می‌کنیم، گرما در صفحه منتشر می‌شود و هر نقطه گرما را از نقاط همسایه خود با ضریب C دریافت کرده و گرم‌تر می‌شود. برنامه‌ای بنویسید که به صورت چند ریسمانی پخش شدن گرما در یک صفحه را نمایش دهد، برای محاسبه گرمای هر نقطه از گرمای نقاط همسایه آن استفاده کنید. هر ریسمان می‌بایست دمای بخشی از صفحه را بروزرسانی کند. نقطه‌ای که منبع گرما در آن قرار می‌گیرد، ضریب C و اندازه صفحه را از ورودی دریافت کنید و پخش شدن گرما را در خروجی نمایش دهید. برای آسان شدن محاسبه دمای نقاط مختلف صفحه، می‌توانید صفحه را یک آرایه دو بعدی $n \times n$ فرض کنید که n در ورودی وارد می‌شود. تعداد ریسمان‌ها را در ابتدا ۱ قرار داده (بصورت سری این کار انجام شود) و تعداد ریسمان‌ها را زیاد کنید و بهترین تعداد را برای ریسمان‌ها گزارش کنید. برای نمایش بهتر نحوه پخش شدن در ریسمان‌ها تابع sleep را صدا بزنید.

۲۸. مسئله موج

فرض کنید یک طناب کشسانی داریم که یک سر آن به یک وسیله متصل است که آن وسیله در آن موج سینوسی ایجاد می‌کند و سر دیگر آن نیز می‌تواند نوسان کند. با وارد شدن موج سینوسی توسط منبع موج هر نقطه به نقطه‌ای که همسایه چپ آن در راستای عمود قرار دارد جا به جا شده و به این ترتیب موج در طول طناب جا به جا می‌شود. برنامه‌ای بنویسید که به صورت چند ریسمانی پخش شدن موج بر روی یک طناب را نمایش دهد، هر ریسمان می‌بایست ارتفاع بخشی از طناب را بروزرسانی کند. برای آسان شدن محاسبه ارتفاع نقاط مختلف طناب، طناب را به n قسمت تقسیم کرده و آن را یک آرایه یک بعدی n تایی فرض کنید که n در ورودی وارد می‌شود. نحوه پخش شدن موج روی طناب را در خروجی نمایش دهید. تعداد ریسمان‌ها را در ابتدا ۱ قرار داده (بصورت سری این کار انجام شود) و تعداد ریسمان‌ها را زیاد

کنید و بهترین تعداد را برای ریسمان‌ها گزارش کنید. برای نمایش بهتر نحوه پخش شدن در ریسمان‌ها تابع sleep را صدا بزنید.

۲۹. پیدا کردن یک عکس در یک عکس دیگر

هر عکس یک ماتریس $n \times m$ است، برای پیدا کردن یک عکس در عکس دیگر باید عکس کوچکتر را در عکس بزرگتر حرکت داد و در هر نقطه ای که عکس کوچک تر را قرار می‌دهیم تمامی پیکسل‌های آنرا با پیکسل‌های متناظر در عکس بزرگتر مقایسه کنیم.

برنامه ای بنویسید که عکس بزرگ و کوچک را ورودی گرفته و به صورت چندریسمانی عکس کوچک تر را بر روی نقاط مختلف عکس بزرگ تر قرار دهد و وجود داشتن یا نداشتن عکس کوچک در عکس بزرگ را در خروجی چاپ کند. اگر عکس وجود داشت مختصات آن نیز در خروجی چاپ شود.

۳۰. محاسبه هیستوگرام یک تصویر

هر عکس سیاه و سفید یک ماتریس $n \times m$ است که هر عنصر آن شدت روشنایی آن نقطه را نمایش می‌دهد. هیستوگرام یک تصویر به نموداری گفته می‌شود که تعداد تکرار هر مقدار از شدت روشنایی در یک تصویر را نمایش می‌دهد. در این نمودار راستای y تعداد تکرار هر مقدار را نشان می‌دهد و راستای x مقادیر قابل قبول را نمایش می‌دهد (مقادیر شدت روشنایی از ۰ تا ۲۵۵ است).

برای مثال مقدار ستون ۱ نشان دهنده این است که عدد یک در این ماتریس چند بار تکرار شده است. برنامه‌ای بنویسید که به صورت چندریسمانی هیستوگرام یک تصویر را بسازد. بدین منظور ابتدا عکس را به چند قسمت تقسیم کرده و هر کدام را برای محاسبه محدوده خود برنامه‌ریزی کنید. سپس این هیستوگرام‌ها را با هم جمع بزنید و در خروجی نمایش دهید. تعداد بهینه برای این عمل چند ریسمان است؟ سرعت محاسبه شدن خروجی را با حالتی که هیستوگرام با یک ریسمان محاسبه شود، مقایسه کنید.

C treats you like a consenting adult. Pascal treats you like a naughty child. Ada treats you like a criminal.
“Bruce Powel Douglass”
