

دانشگاه علم و صنعت

دانشکده مهندسی کامپیوتر

درجه تحصیلی: کارشناسی

گزارشکار تکلیف2 OS

گردآورنده:

پرنیان شاکریان - 99400064

استاد:

دکتر انتظاری

سال تحصیلی: اسفند 1401

خلاصه:

در تکلیف دوم سیستم عامل قصد داریم از Process و Thread در حل تمارین استفاده کنیم. اگر بخواهیم به طور خلاصه در مورد ماهیت کدها توضیح دهیم Process، پردازه مجموعه‌ای از کدها، حافظه، داده و سایر منابع است. Thread توالی از کدها به حساب می‌آید که در داخل محدوده یک پردازه اجرا می‌شود. هر رشته می تواند بخشی از یک وظیفه را اجرا کند (به عنوان مثال، جمع کردن عناصر یک آرایه)، یا می تواند همان کار را برای client مختلف در client-server اختصاص دهد.

شرح کلی:

1. اضافه کردن کتابخانه‌ها(Include the header file)
2. هر رشته دارای یک شی از نوع pthread\_t مرتبط با آن است که شناسه آن را می گوید. (نمی تواند توسط چندین رشته به طور همزمان استفاده شود)
3. یک رشته ایجاد می شود و با استفاده از تابع ()pthread\_create شروع می شود و نیازمند چهار پارامتر است (id, Attributes, Starting routine, Arguments)
4. () pthread\_exit برای خروج از thread استفاده می شود.(معمولاً در پایان روتین شروع نوشته می شود.) اگر مقداری پس از پایان توسط یک رشته برگردانده شود، مرجع آن به عنوان یک آرگومان ارسال می شود.
5. parent با استفاده از ()pthread\_join منتظر یک childمی‌شود. دو پارامتر این تابع عبارتند از Thread ID و Reference to return value

نکته: (نوع برگشتی شروع و آرگومان آن معمولاً بر روی void\* تنظیم می شود.)

تمام کتابخانه ها و headerهای مورد نیاز ما برای تمرین 2 به طور کلی و برای آسانی کار اینجا نوشته و اشاره شده است.

#include <assert.h>

#include <stdio.h>

#include <pthread.h>

#include <stdlib.h>

#include <unistd.h>

#include <sys/types.h>

#include <sys/stat.h>

#include <sys/wait.h>

#include <string.h>

#include "stdlib.h"

#include "stdio.h"

#include "unistd.h"

#include "string.h"

#include "pthread.h"

برای ساخت makefile ها از لینک های زیر کمک گرفته ام.

<https://youtu.be/bPMDEdjKBWA>

<https://youtu.be/_r7i5X0rXJk>

سوال 2: برنامه ای بنویسید که با استفاده از N تا Thread در N فایل مختلف 100 عدد رندوم جنریت و رایت کند و سپس با استفاده از N تا Thread دیگر اون N\*100 عدد را بخواند و میانگین آن ها را محاسبه کرده و چاپ کند.

در ابتدا کتابخانه‌ها و headerهای مورد نیاز خود را include خواهیم کرد. (کتابخانه های مورد نیاز ما برای این تمرین در اول داک نوشته شده است) در حالت کلی این برنامه چهار فایل حاوی 100 تصادفی تولید می کند( تعداد اعداد را میتوانیم خودمان تعیین کنیم) و سپس میانگین آن اعداد را محاسبه می کند.

در ابتدا ساختاری به نام custom\_data را تعریف می کنیم که دارای دو متغیر thread\_num (یک عدد صحیح) و thread\_avg است. متغیر thread\_num برای شناسایی فایلی که thread روی آن کار می کند و متغیر thread\_avg برای ذخیره میانگین اعداد در آن فایل استفاده می شود. تابع main برخی از متغیرها را مقداردهی اولیه می کند. با ایجاد (gen\_threads) برای تولید فایل های تصادفی شروع می شود. سپس منتظر می ماند تا با استفاده از تابع ()pthread\_join تولید را به پایان برسانند. پس از تکمیل رشته جدید (calc\_threads) برای محاسبه میانگین فایل های تولید شده ایجاد می شود. آرایه ای از ساختارهای calc\_thread\_args ایجاد می کند تا تعداد رشته و میانگینی را که هر رشته محاسباتی محاسبه می کند ذخیره کند. هر رشته یک آرگومان عدد صحیح (thread\_num) می گیرد که مشخص می کند روی کدام فایل باید عمل کند. سپس 100 عدد تصادفی (Num\_lines) بین 1 تا 100000 تولید می‌کند، آنها را در فایل نوشته و خارج می‌شود. هر رشته محاسباتی یک آرگومان ساختار custom\_data را می گیرد که مشخص می کند روی کدام فایل باید عمل کند سپس فایل را باز کرده، اعداد را می خواند و آنها را جمع می کند، میانگین را محاسبه می کند و آن را در متغیر thread\_avg ساختار custom\_data ذخیره می کند.

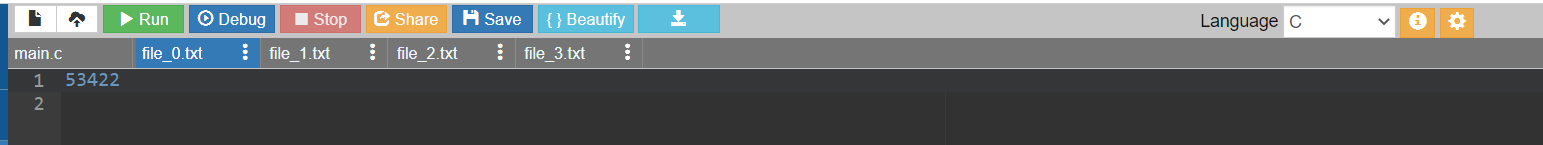
پس از تکمیل همه محاسبات، تابع main میانگین‌های هر رشته را جمع‌ می‌کند و بر تعداد رشته‌ها تقسیم می‌کند تا میانگین نهایی همه فایل‌های تولید شده را بدست آورد. این میانگین روی کنسول چاپ می شود. در نهایت، برنامه خارج می شود.

به طور خلاصه برنامه از 3 قسمت اصلی تشکیل شده است:

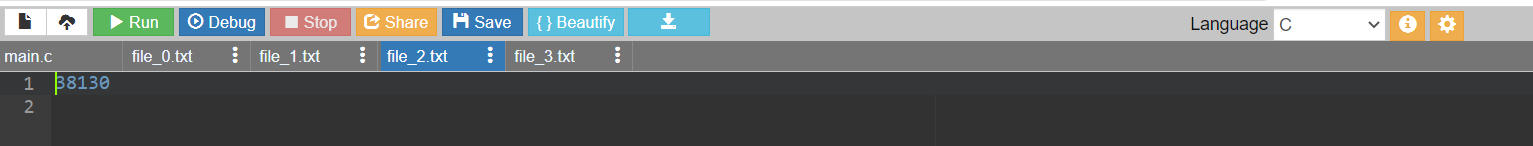
1. قسمت اول تابعی به نام ()gene\_random\_files را برای تولید آرایه ای از اعداد تصادفی تعریف می کند و آنها را بر اساس شماره رشته در فایلی به نام file\_{thread\_num}.txt می نویسد.
2. در قسمت دوم تابع ()calault\_average را تعریف می کنیم که اعداد هر فایلی را که قبلاً ایجاد شده بود بر اساس شماره رشته خوانده و میانگین آنها را محاسبه می کند. پارامترهای لازم برای محاسبه میانگین را در ساختار "custom\_data" ذخیره می کند.
3. قسمت اخر عملکرد اصلی برنامه است. دو آرایه رشته ایجاد می کند، یکی برای تولید اعداد تصادفی و نوشتن آنها در فایل ها و دومی برای محاسبه میانگین اعداد برای هر فایل. سپس منتظر می ماند تا همه رشته ها کار خود را به پایان برسانند، میانگین نهایی هر چهار رشته را محاسبه می کند و آن را در کنسول چاپ می کند.

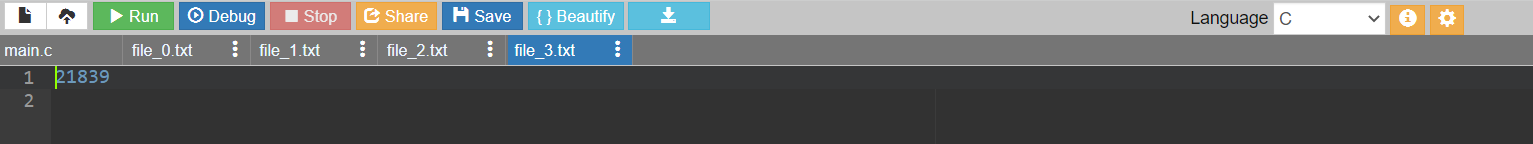
وقتی برنامه را ران کنیم چنین خروجی را به ما خواهد داد:

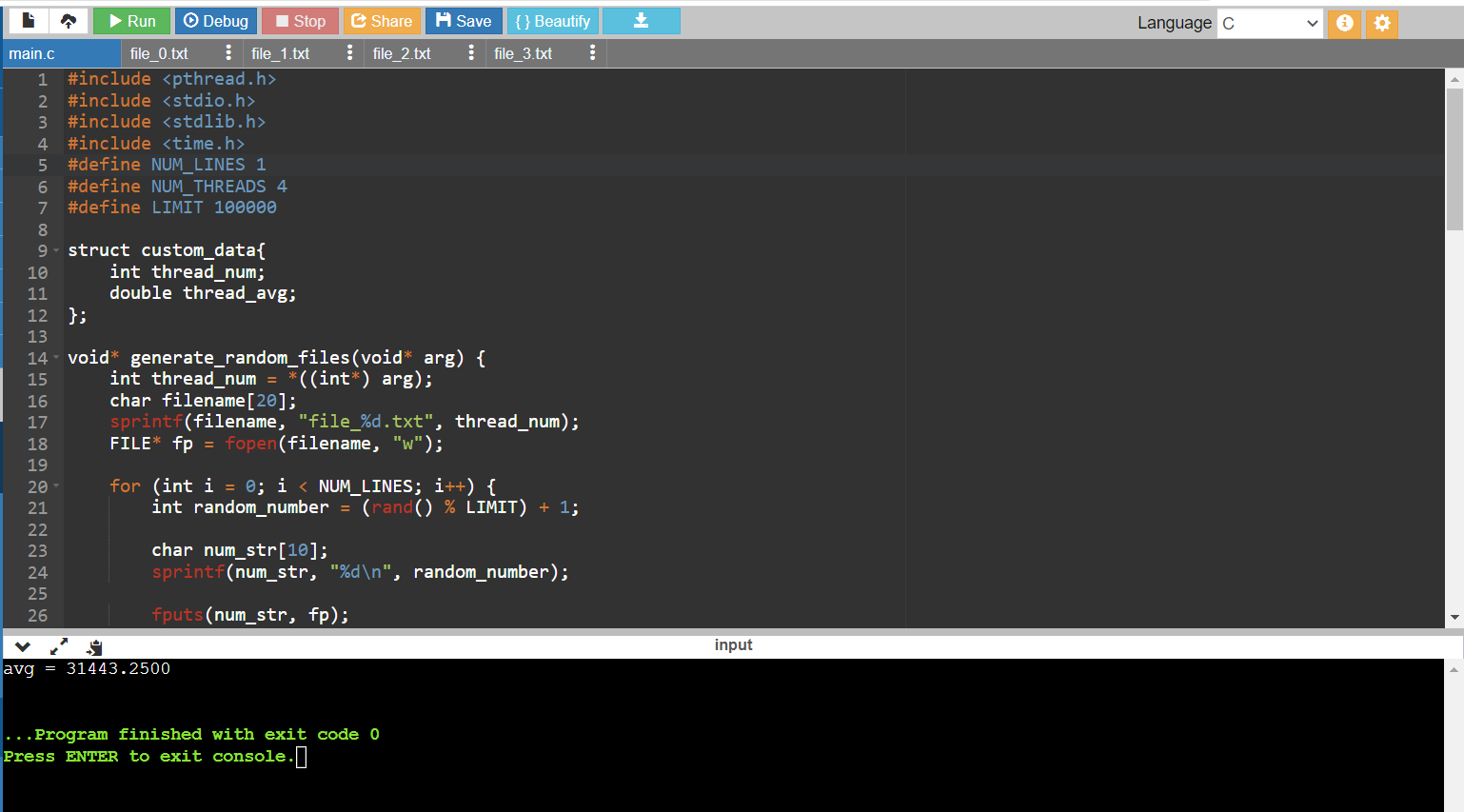
(توجه برای راحتی تست num را به جای ,100 1 گذاشته ام)











اگر اعداد داخل هر فایل رو دستی جمع کرده و سپس میانگین بگیریم جواب 31443.25 همان avg که خودش حساب میکند میشود.