

شرح پروژه

هدف از این پروژه طراحی و پیاده‌سازی یک برنامه کاربردی Multi-thread مشابه با ابزار grep در لینوکس است. این ابزار برای جستجوی کلمات در فایل‌های متنی استفاده می‌شود. شما باید با استفاده از زبان C یک ابزار درست کنید که بتواند به صورت recursive در یک پوشه، در تمامی فایل‌ها به دنبال یک عبارت string بگردد و تمامی نتایج را لیست کند و همچنین تعداد match ها را خروجی دهد.

برخلاف ابزار grep نیاز به پیاده‌سازی قابلیت سرچ با regex نیست و بیشتر تمرکز بر multi-thread بودن این ابزار است. توصیه می‌شود قبل از شروع برنامه یک بار برنامه grep را امتحان کنید تا درک بهتری از طرز کار آن بدست آورید. برای اینکار می‌توانید در یک توزیع لینوکسی این برنامه را امتحان کنید.

مفاهیم پروژه

ایجاد فرآیند

با ایجاد یک فرآیند والد شروع کنید و پس از آن برای هر زیر پوشه در فهرست اولیه، یک فرآیند ایجاد کنید.

مدیریت thread

هر یک از این فرآیندهای فرزند باید برای هر فایل در زیر پوشه‌ی تعیین‌شده خود، thread ایجاد کنند.

مکانیسم‌های همگام‌سازی

از قفل‌ها و سمافورها برای رسیدگی به مشکلات همگام‌سازی که ممکن است به دلیل داده‌های مشترک ایجاد شود، استفاده کنید.

ارتباطات بین‌فرآیندی

در این بخش شما باید مکانیسم‌های ارتباط بین‌فرآیندی (IPC) مانند لوله‌ها (Pipes)، صف‌های پیام (Message Queue) و حافظه مشترک (Shared Memory) را پیاده‌سازی کنید. با این کار می‌توان فرآیندهای فرزند را قادر ساخت تا داده‌های مورد نظر خود را به فرآیند والدی که آن‌ها را ایجاد کرده بازگردانند.

پس به صورت خلاصه برای بررسی هر پوشه شما بایستی یک process و برای بررسی هر فایل یک thread توسط process مربوطه بسازید. و در نهایت داده خروجی thread ها را در process مربوطه پردازش و جمع‌آوری بکنید. و به همین صورت داده‌های process ها را به شکل recursive به process پدر انتقال دهید تا در نهایت تمامی داده‌ها به دست main process برسد.

مواردی که برنامه‌ی شما باید خروجی دهد.

1. تعداد کل فایل‌های بررسی شده
 2. تعداد match عبارت ورودی
 3. لیستی از آدرس‌های هر match با این فرمت در هر خط: (file_address:match_line:match_character)
برای مثال اگر کلمه hello در ۲ فایل پیدا شود:
- ```
./directory1/File1.txt:60:1
./directory2/directory3/File2.txt:12:10
```
- که یعنی کلمه hello یک بار در فایل File1.txt و در خط ۶۰ و در کاراکتر ۱ شروع شده است. و همچنین یک بار در فایل File2.txt خط ۱۲ و کاراکتر ۱۰م پیدا شده است.

## فرمت ورودی و خروجی

### ورودی

می‌توانید برنامه را به گونه‌ای بنویسید که عملیات را دقیقاً در همان مسیری که در حال اجراست شروع کند و یا مسیر را به عنوان ورودی به آن بدهیم:

1- طرز استفاده از ابزار باید به این شکل در محیط cli باشد:

```
user@local:~# ./mygrep ./path_to_directory
```

**نکته:** دقت کنید که برنامه شما باید به صورت recursive عمل کند و به عنوان ورودی آدرس directory می‌گیرد. و بایستی تمامی فایل‌های داخل دایرکتوری را لیست کند.

2. پس از اینکه مسیر اجرا برای برنامه مشخص شد و اجرا پایان یافت می‌تواند خروجی مشابه نمونه زیر نمایش دهد:

```
Total number of files: 1023
Total number of Matches: 2
./directory1/File1.txt:60:1
./directory2/directory3/File2.txt:12:10
```

**نکته:** الزاماً نباید خروجی شما به این شکل باشد و می‌توانید بر اساس سلیقه‌ی خود، خروجی آن را تغییر دهید و یا حتی خروجی گرافیکی را نمایش دهید.

توجه داشته باشید برنامه‌ی grep برنامه‌ای پیشرفته و حرفه‌ای است و نیازی نیست برنامه‌ی شما کاری مشابه آن بکند؛ بلکه هدف از توصیه این برنامه این است که شما بهتر بتوانید روال شروع از یک پوشه توسط این برنامه را ببینید تا درک بهتری از صورت پروژه به دست آورید.

## نمره اضافه

### رابط گرافیکی

در رابط گرافیکی شما باید ابتدا یک فولدر از سیستم را به صورت گرافیکی انتخاب کنید (File Select) و سپس داده‌های مدنظر را نمایش دهد.

### قابلیت‌های debug

نام thread و داده‌هایی که توسط آن capture شده‌اند (به عنوان مثال هر thread کدام فایل‌ها را بررسی کرده است). و همچنین مقدار زمانی که صرف پردازش هریک از thread ها شده است.

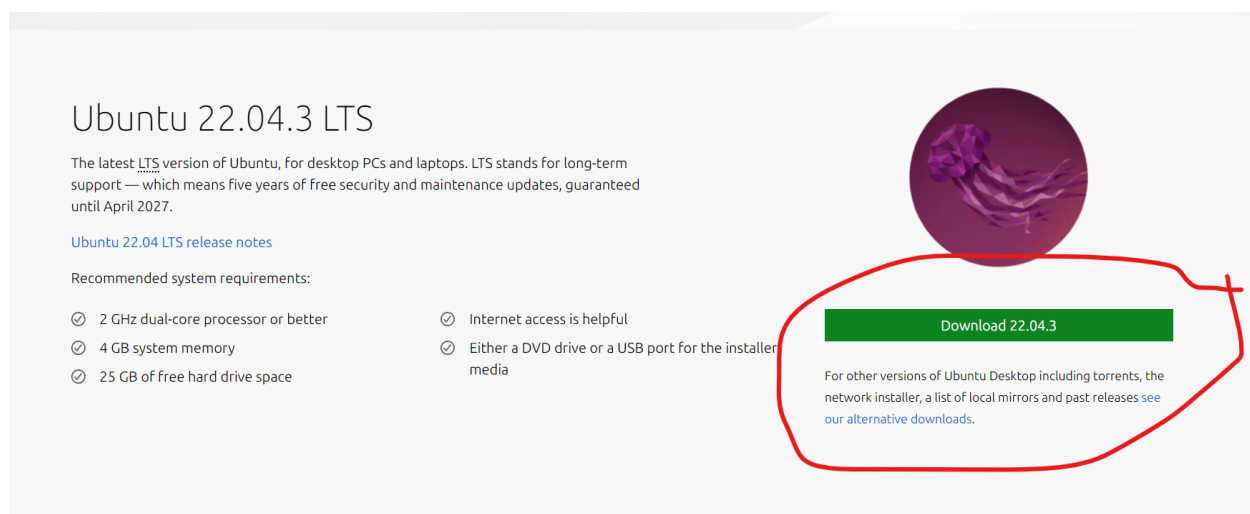
### قابلیت سرچ regex

امکان جستجو با استفاده از عبارات regex. برای پیاده‌سازی این قسمت استفاده از هر کتابخانه‌ای بلامانع است.

### استفاده از thread pool یا process pool

شما می‌توانید به جای اینکه به ازای هر فایل یک thread درست کنید، می‌توانید از یک thread pool استفاده کنید و بعد از تمام شدن کار هر thread آن را دور نیندازید و دوباره از آن thread بهره بگیرید.

حتماً توصیه میشود از یک توزیع لینوکسی برای محیط توسعه خود استفاده کنید. اگر از ویندوز استفاده می‌کنید می‌توانید از VirtualBox استفاده کنید تا یک توزیع لینوکسی را در یک ماشین مجازی استفاده کنید. یک توزیع لینوکسی که می‌توانید امتحان کنید Ubuntu است. [لینک دانلود](#)



### راه دیگر به جز نصب توزیع لینوکسی:

یک روش دیگر برای ایجاد محیط توسعه در ویندوز استفاده از [msys2](#) است. با استفاده از این برنامه می‌توانید از کتابخانه‌های استاندارد POSIX زبان C در ویندوز استفاده کنید و برنامه‌ها خود را کامپایل کنید. این روش توصیه نمی‌شود اما اگر محدودیت در نصب و دانلود یک توزیع لینوکسی دارید می‌توانید از این روش کارراه‌انداز استفاده کنید.

همچنین یک راه دیگر نصب لینوکس روی ویندوز استفاده از WSL2 در ویندوز است. در این مورد می‌توانید در اینترنت جستجو کنید.

### توضیحات تکمیلی

- 1) پروژه به صورت *انفرادی* یا در *گروه‌های 2 نفره* قابل انجام است (آپلود توسط هر دو عضو الزامی‌ست).  
**گروه‌ها می‌بایست از گیت استفاده کنند.**
- 2) زبان پیاده‌سازی پروژه، فقط زبان برنامه‌نویسی C می‌باشد (برای پیاده‌سازی رابط گرافیکی، محدودیت زبانی ندارید).
- 3) مراحل پیاده‌سازی و نحوه‌ی اجرای برنامه‌ی خود را حتماً در فایل `readme.md` به صورت کامل توضیح دهید.
- 4) هنگام تحویل، هر دو عضو گروه باید تسلط کامل داشته باشند.
- 5) فایل نهایی (شامل کد و فایل `readme.md`) را به فرمت `"FileMonitoring_<Student.IDs>_<Student.names>.zip"` در Vu بارگذاری کنید.
- 6) در صورت مشاهده هرگونه شباهت میان گروه‌ها، **نمره 100-** به هر دو گروه داده می‌شود.

**مهلت تحویل:** 1 بهمن ماه 1402 خورشیدی.

"موفق باشید"