دستور کار جلسه سوم

۱. همه موارد خواسته شده در این سؤال را با استفاده از system call ها بنویسید.

می‌خواهیم یک برنامه c بنویسیم که می‌تواند آرگومان‌هایی مطابق جدول زیر داشته باشد و عملیات مربوطه را انجام دهد:

-c path permission	فایلی در مسیر path ساخته و سطح دسترسی permission را برای آن تعیین می‌کند. permission به صورت octal دریافت می‌شود. دقت کنید اگر فایل از قبل وجود داشت فایل جدید جایگزین قبلی شود.
-t path num	چک میکند که آیا فایلی که در مسیر path قرار دارد دارای سطح دسترسی num (عدد اوکتال یک رقمی) برای پروسس فعلی هست یا خیر (خط اول خروجی) و اگر جواب خیر بود در سطر دوم خروجی، سطح دسترسی را به صورت عدد اوکتال یک رقمی مینویسد.
-s path	اطلاعات مربوط به فایلی که در مسیر path قرار دارد در سه خط نمایش داده می شود. خط اول: UserID مالک فایل خط دوم: آخرین تایم تغییر (Modify) فایل خط سوم: سایز فایل (بایت)
-m dirPath prefix ext v1 v2	در دایرکتوری با آدرس dirPath (فرض میشود که وجود دارد و نیاز به ساخت آن نیست)، فایل هایی با نام prefix_idx.ext ایجاد می کند که در آن idx مقداریست بین v1 و v2. سطح دسترسی این فایل‌ها 755 تنظیم شود.

برای پیاده‌سازی این برنامه حتما باید از روش کتابخانه استاتیک استفاده کنید. به این صورت که برای توسعه کتابخانه یک فایل h و c (مشابه آزمایشگاه هفته پیش) ساخته و برای هر یک از عملیات‌های ذکر شده یک تابع پیاده‌سازی کنید. به طور خلاصه، کتابخانه شما باید حداقل شامل توابع زیر باشد:

```
void createFileWithPermission(const char * path, int permission)

int checkFile(const char * path, int permission)

struct mystruct showFileInfo(const char * path)

void createFileList(const char * dirPath, const char * prefix, const char * ext, int from, int to)
```

برنامه اصلی (main.c) شما باید از توابع کتابخانه شما برای پیاده‌سازی عملکردهای خواسته شده در جدول اول استفاده کند.

نکته نیازی به بررسی درست بودن آرگومان‌های ورودی توسط کاربر در برنامه نیست، فرض می‌کنیم کاربر برنامه را به شکل صحیح اجرا می‌کند.

نکته نیازی به ساخت دایرکتوری‌های والد موجود در آرگومان path نیست.

نکته از تابع ctime برای نمایش زمان و از تابع strtol برای تبدیل رشته به عدد صحیح استفاده کنید. (حتما man page توابع خوانده شود)

نکته برای ساخت کتابخانه و کامپایل برنامه، حتما از makefile استفاده کنید، فایل میک شما باید حداقل دارای تارگت‌های زیر باشد

```
all:
تارگت دیفالت
compile:
فایل آبجکت اصلی و فایل‌های کتابخانه را دریافت کرده و کامپایل نهایی را انجام می‌دهد
main.o:
ساخت آبجکت فایل اصلی
libmylibrary.a:
ساخت فایل استاتیک لایبرری
mylibrary.o:
ساخت فایل آبجکت
clean:
پاک سازی دایرکتوری از تمامی فایل‌های اضافه آبجکت و لایبرری
```

حتما dependency ها را در تارگت‌ها رعایت کنید

مثال ۱) :

در مسیر /home/share/shared/ فایلی با نام 1.txt وجود دارد:

```
./app -s /home/share/shared/1.txt
-------------------------
result:
1000
Mon Aug 13 08:23:14 2022
2
```

مثال ۲):

هدف اجرا ایجاد فایلهایی با پیشوند drawing، پسوند img و اندیس هایی از 1 تا 5 در مسیر /home/testDir/ می باشد:

```
./app -m /home/testDir/  drawing  img  1  5
-------------------------
result:
/home/testDir/drawing_1.img
/home/testDir/drawing_2.img
/home/testDir/drawing_3.img
/home/testDir/drawing_4.img
/home/testDir/drawing_5.img
```