



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.
-w path content	برنامه محتوای content را در فایلی در مسیر path می‌نویسد. در صورتی که در مسیر path فایلی وجود نداشته باشد، فایل جدید ایجاد شده و سطح دسترسی 755 برای آن تعیین می‌شود.
-r path	محتوای فایلی که در مسیر path قرار دارد نمایش داده می‌شود.
-m dirPath prefix ext v1 v2	در دایرکتوری با آدرس dirPath، فایل‌هایی با نام prefix_idx.ext ایجاد می‌کند 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)
void createFileWithContent(const char * path, const char * content)
void printFileContent(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 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 آbjکت اصلی و فایل‌های کتابخانه را دریافت کرده و کامپایل نهایی را انجام می‌دهد
main.o:	ساخت آbjکت فایل اصلی
libmylibrary.a:	ساخت فایل استاتیک لایبری
mylibrary.o:	ساخت فایل آbjکت
clean:	پاک سازی دایرکتوری از تمامی فایل‌های اضافه آbjکت و لایبری

حتما dependency ها را در تارگت‌ها رعایت کنید

مثال (۱) :

در مسیر /home/share/shared/ فایل‌ی با نام txt.1 وجود دارد که در آن مقدار "this is 1.txt" نوشته شده است:

```
./app -r /home/share/shared/1.txt
-----
result:
this is 1.txt
```

مثال (۲) :

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

```
./app -m /home/testDir/ drawing img 1 5
-----
result:
/home/testDir/drawing1.img
/home/testDir/drawing2.img
/home/testDir/drawing3.img
/home/testDir/drawing4.img
/home/testDir/drawing5.img
```