

Libraries

Common scenario

main.c

math-sin.c

math-cos.c

```
$ gcc main.c math-sin.c math-cos.c
```

Program

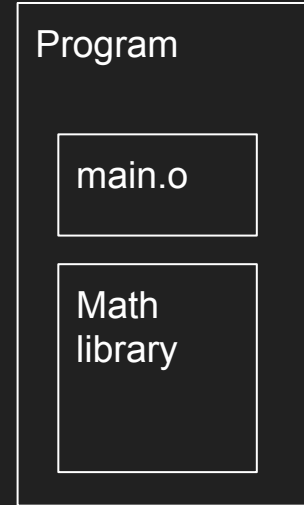
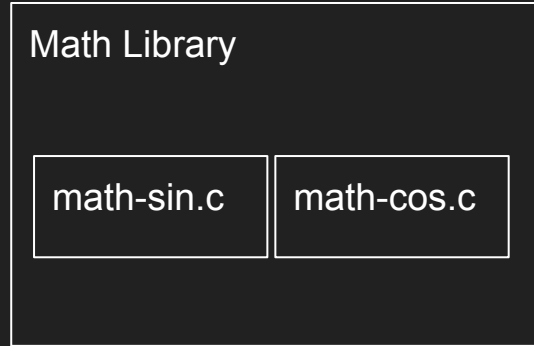
main.o

math-sin.o

math-cos.o

Common scenario - introducing library

main.c



Libraries - archive



```
$ gcc -c math-sin.c math-cos.c
```

```
$ ar rcv libsincos.a math-sin.o math-cos.o
```

`libsincos.a` now contains all objects from the two c files

Libraries - archive



```
$ gcc main.c -L. -lsincos -o main
```

`-L.` look for libraries in `.`
`-lsincos` link a library called `libsincos`

`main` contains all objects from `main.c` and the library

Libraries - archive



```
$ gcc prog.c -L. -lsincos
```

prog contains all objects from main.c and the library

Libraries - archive

main

prog

Both programs contain the objects from the library. What if 20 programs use this library:

- recompile all programs if updating the library?
- disk usage?

Is it possible to share one library and link it “on the fly” instead of linking it statically to the program?

Shared object file



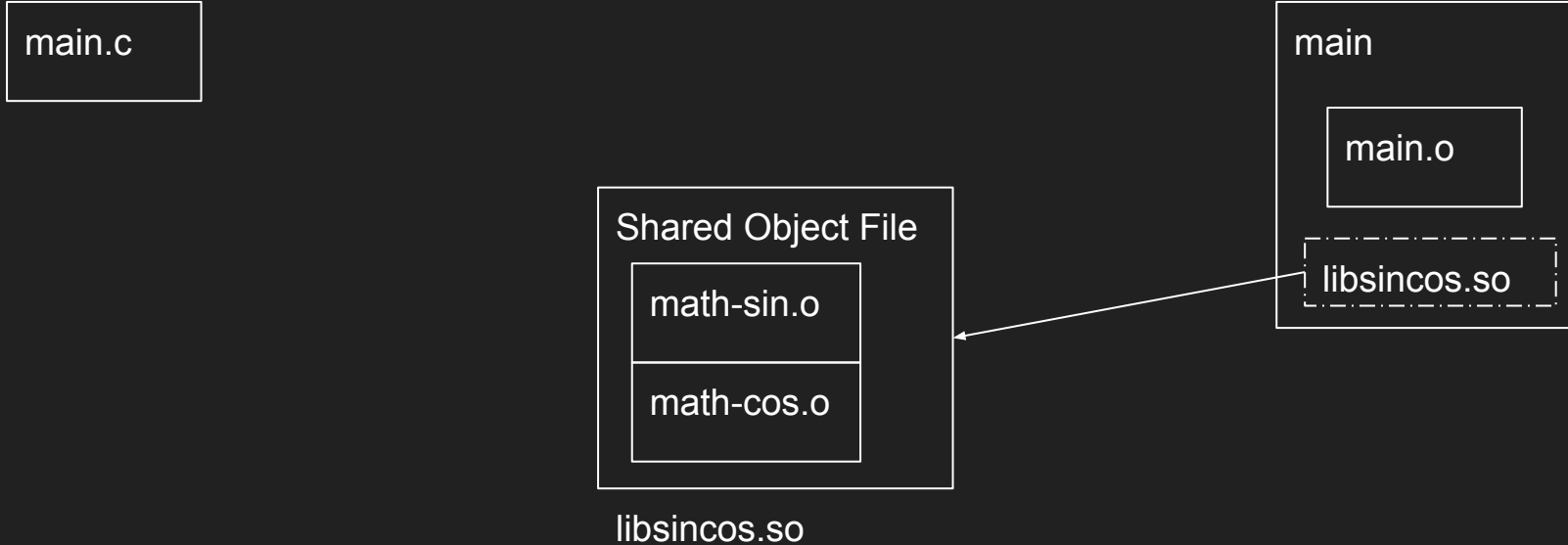
```
$ gcc -shared math-sin.c math-cos.c -o libsincos.so
```


Shared object file



```
$ gcc main.c -L. -lsincos -o main
```

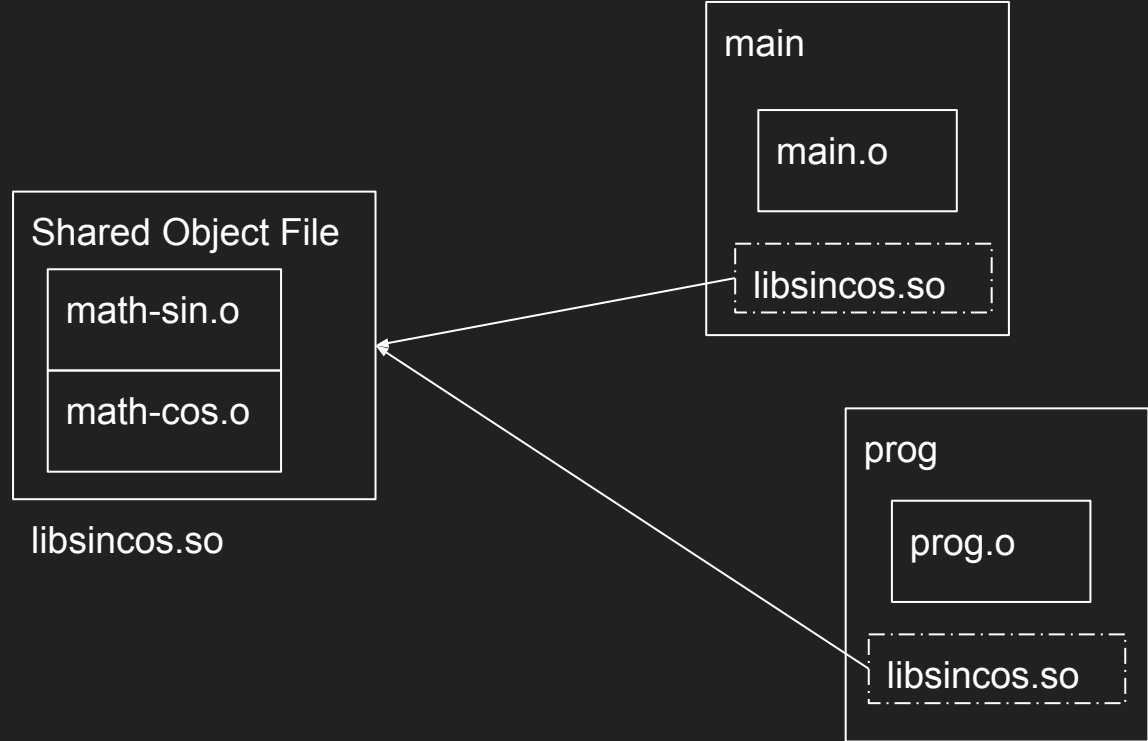
Shared object file



Shared object file

main.c

prog.c



Libraries and header files

When providing a library (static or shared) you (most likely) need to provide corresponding header files.

- `libsincos.so`
- `math-sin.h`
- `math-cos.h`