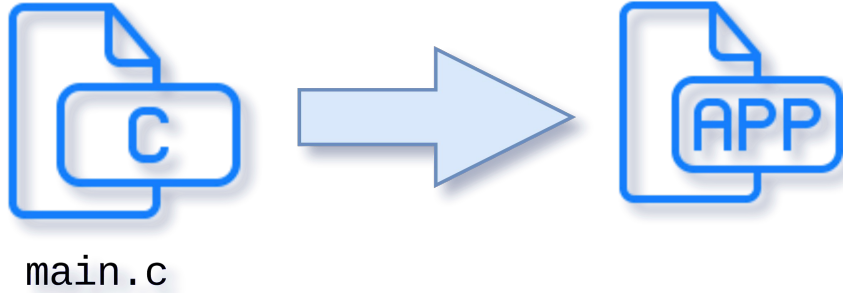


**make**

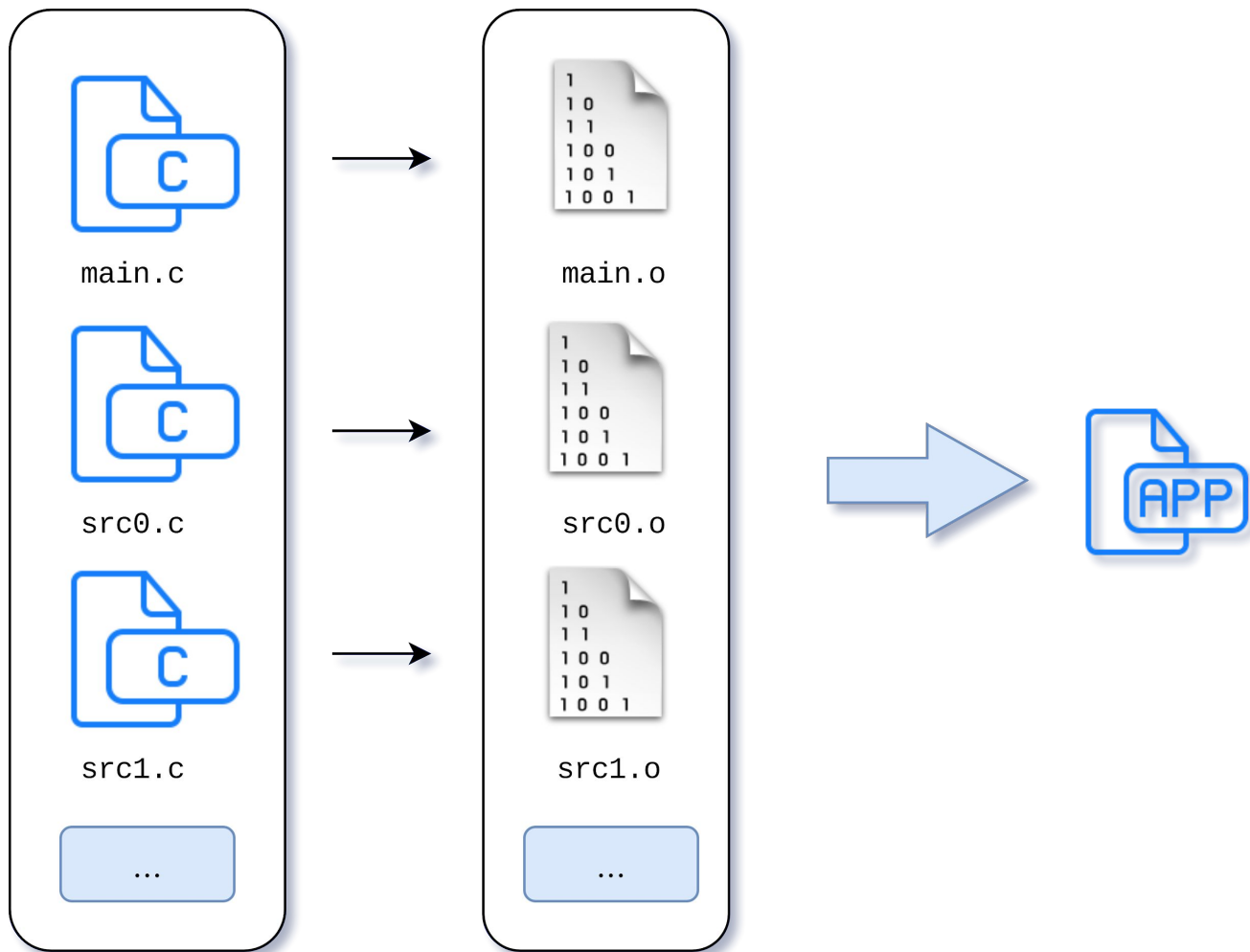
- Что такое make?
- Какие проблемы он решает?

Цель - скомпилировать проект, получив на выходе исполняемый файл.





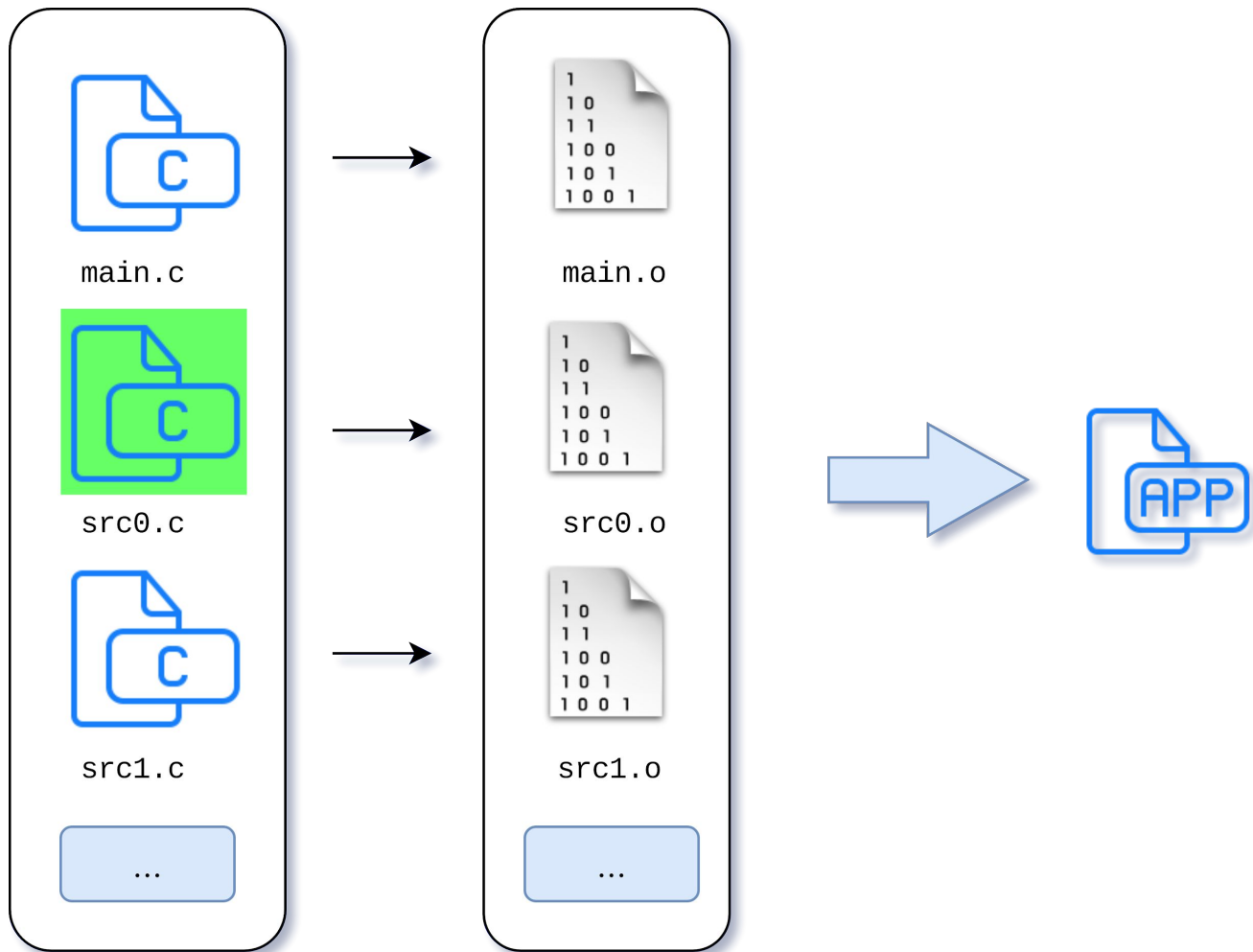
```
1 | $ gcc main.c -o main
```



```
1 $ gcc -c src0.c
2 $ gcc -c src1.c
3 $ gcc -c main.c
4 $ gcc -o main main.o src0.o src1.o
```

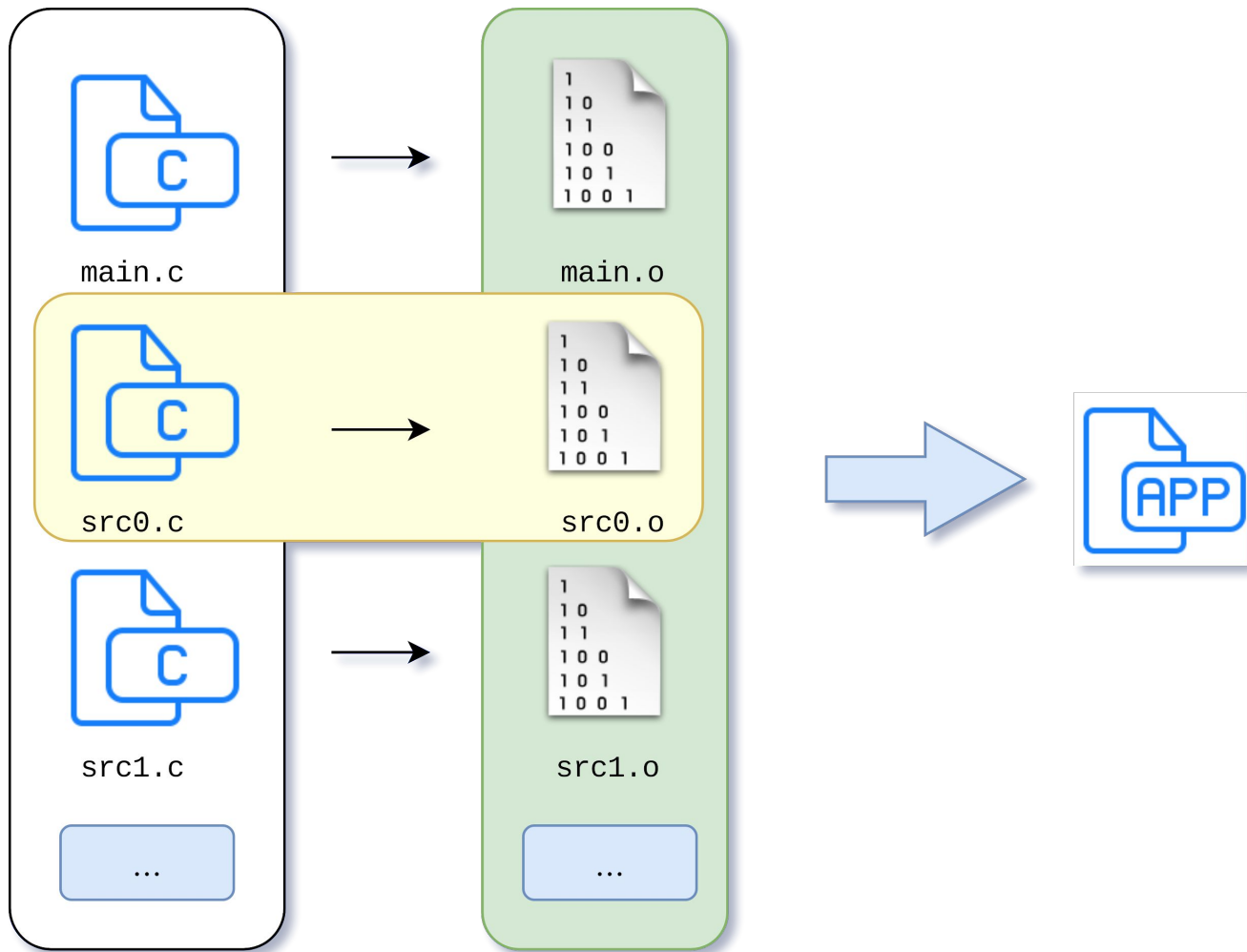
Так проще!

```
1 | $ ./build.sh
```

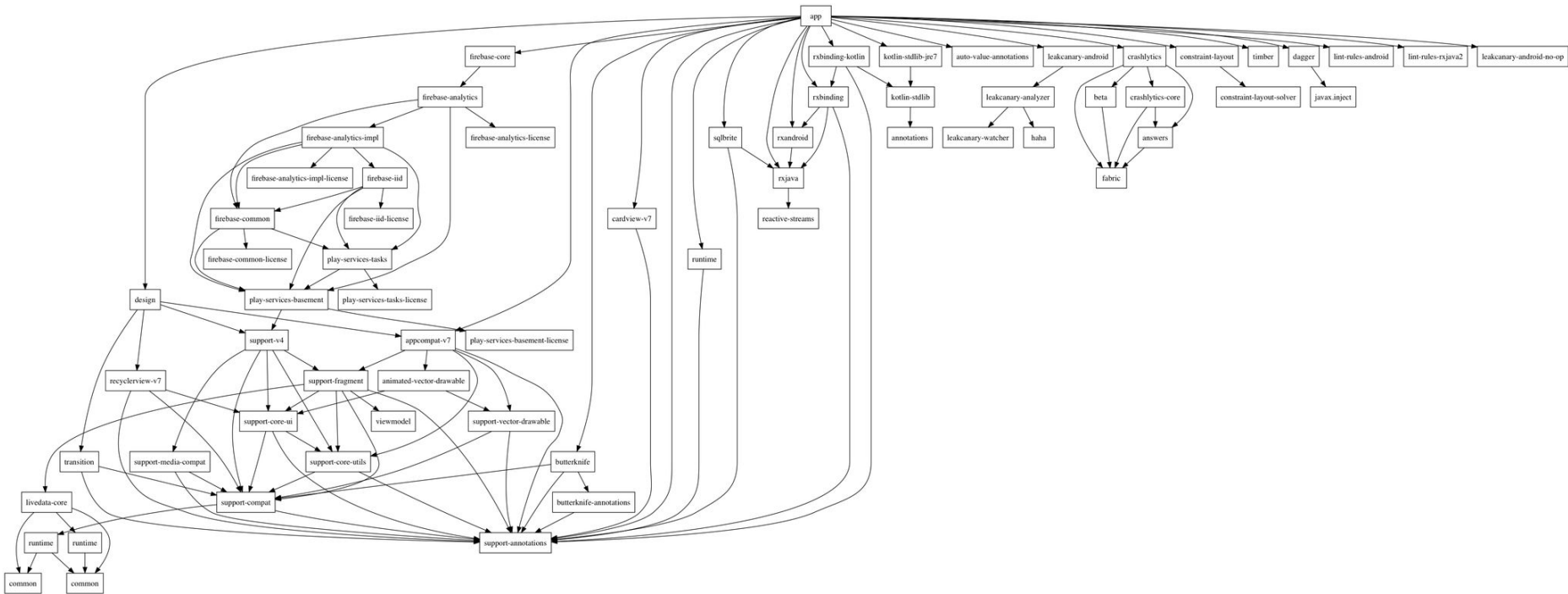




```
1  #!/bin/bash
2
3  function modification_time
4  {
5      date -r "$1" '+%s'
6  }
7
8  function check_time
9  {
10     local name=$1
11     [ ! -e "$name.o" ] && return $?
12     [ "$(modification_time "$name.c")" -gt "$(modification_time "$name.o")" ] && return $?
13 }
14
15 check_time src0 && gcc -c src0.c
16 check_time src1 && gcc -c src1.c
17 check_time main && gcc -c main.c
18 gcc -o main main.o src0.o src1.o
```



## Большой и сложный проект



# Что делает make

- анализ временных меток зависимостей и целей
- выполнение минимального объема работы, необходимого для того, чтобы гарантировать актуальность производных файлов

(+ параллельное выполнение команд)

# Makefile

```
< цель ... > : < зависимости ... >  
    < команда >  
    ...  
    ...
```

## Простейший Makefile

```
1  CC=gcc
2  CFLAGS=-c -Wall
3
4  all: app
5
6  foo.o: foo.c foo.h
7      $(CC) $(CFLAGS) foo.c
8
9  bar.o: bar.c bar.h
10     $(CC) $(CFLAGS) bar.c
11
12  main.o: main.c
13     $(CC) $(CFLAGS) main.c
14
15  app: main.o foo.o bar.o
16     $(CC) -o app main.o foo.o bar.o
17
18  clean:
19     rm -f *.o app
20
21  .PHONY: clean
```

# Генерация Makefile'ов

- autotools
- cmake
- qmake
- qbs
- и т.д.

## autotools

```
1 $ ./configure # генерируется Makefile
2 $ make
3 $ make install
```

## QMake

```
1 $ qmake # генерируется Makefile
2 $ make
3 $ make install
```

**Thanks for attention!**