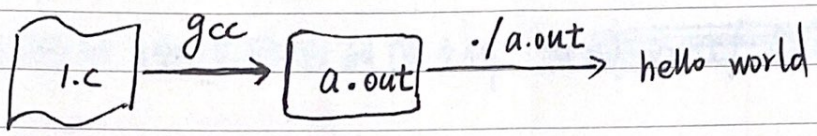
Topic 1: 程序的运行.

code 1: hello world



a.out里是什么? = 进制可执行文件.

如何标识? Magic No.

objdump: 读取分析=进制文件.

问题1: 请写一个程序,运行时打印出main函数的地址。

问题2: 请写一个程序,运行时打印出main函数对应的=进制文件的字节数。

执行命令 ./a.out 时,机器做了些什么?

① 将存在磁盘上的文件读入到内存条. (加载)

存在内存上的什么位置?

② OS找到入口函数main, PC指向它,开始执行.

③ 运行结束后, OS回收内存.

☆: 冯·诺依曼: 程序必须加载到内存才能执行.

问题3: 请写一个程序,程序运行时将自己的可执行文件删除。

为什么要 #include <stdio.h> ?

$ gcc -V 1.c

stdio.h 在哪儿？里面有些什么？

运行时编译器找到库文件，~~找到 printf 函数~~，拷贝出来.

restrict：gcc 扩展.

编译时根据函数声明检查.

code 2：求圆面积：

```
double area( double r){


}
```

定义

area 函数在 main 函数之后定义，有 error. WHY?

C 语言对程序单遍进行扫描

↓

∴ 先声明函数

*关键字 extern

模块化编程：

C语言按照文件区分模块

main.c | 模块1.c 模块1.h | 模块2.c 模块2.h ……

模块

eg： main.c yuan.c yuan.h

编译： $ gcc main.c yuan.c

① 如果在main.c里写 #include "yuan.c" 而不是 "yuan.h"

编译时会发生什么？为什么？

② $ gcc -c yuan.c

生成了一个文件：yuan.o

$ rm -rf yuan.c

删除 yuan.c

$ gcc main.c yuan.o (链接).

编译.

③ 如果在main.c里写入 #include "yuan.h"
#include "yuan.h"

会有什么影响？为什么？