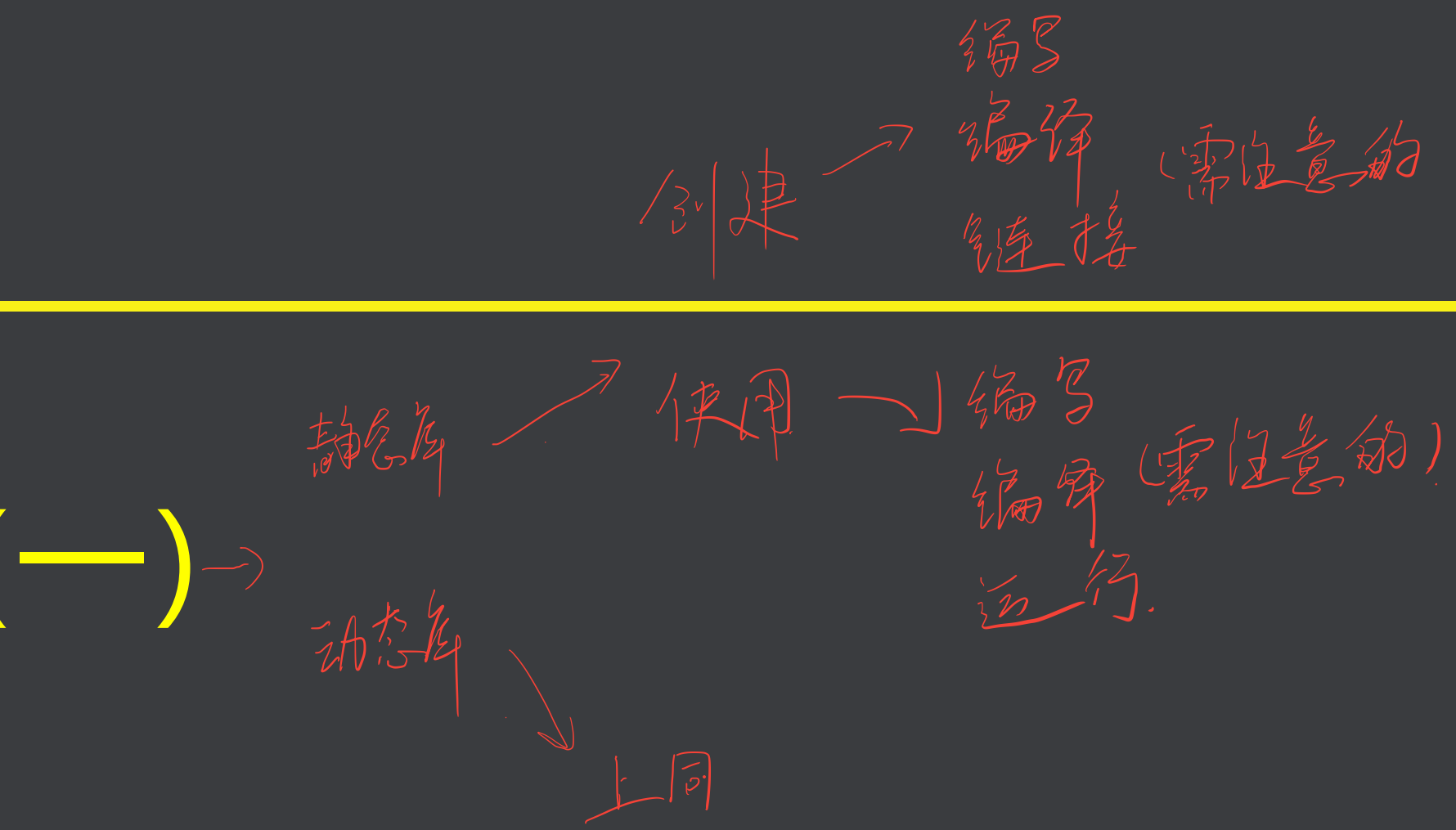
# 程序库（一）




主讲：大海老师

# 课程目标：

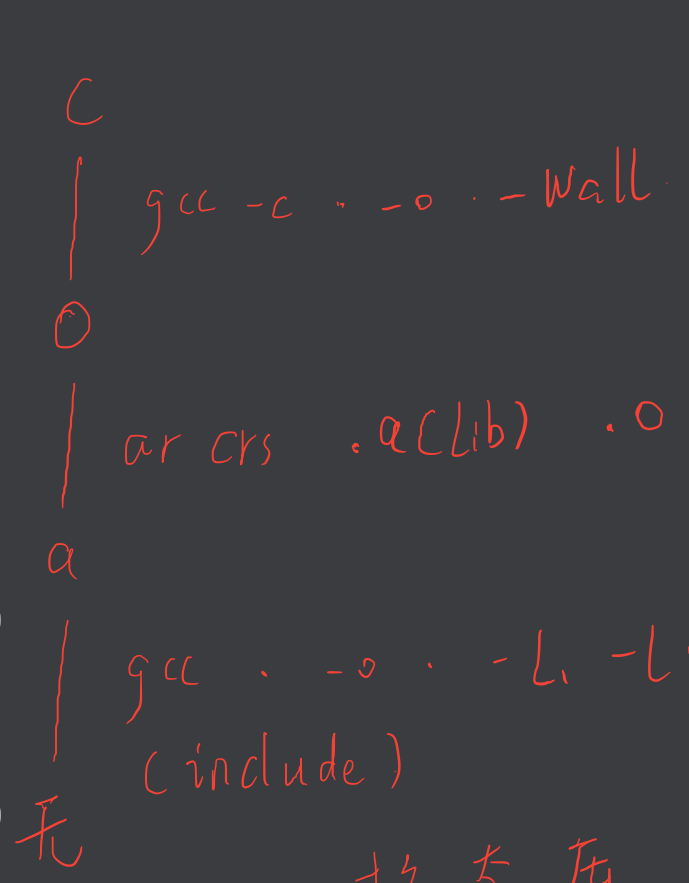
库的概念（了解）

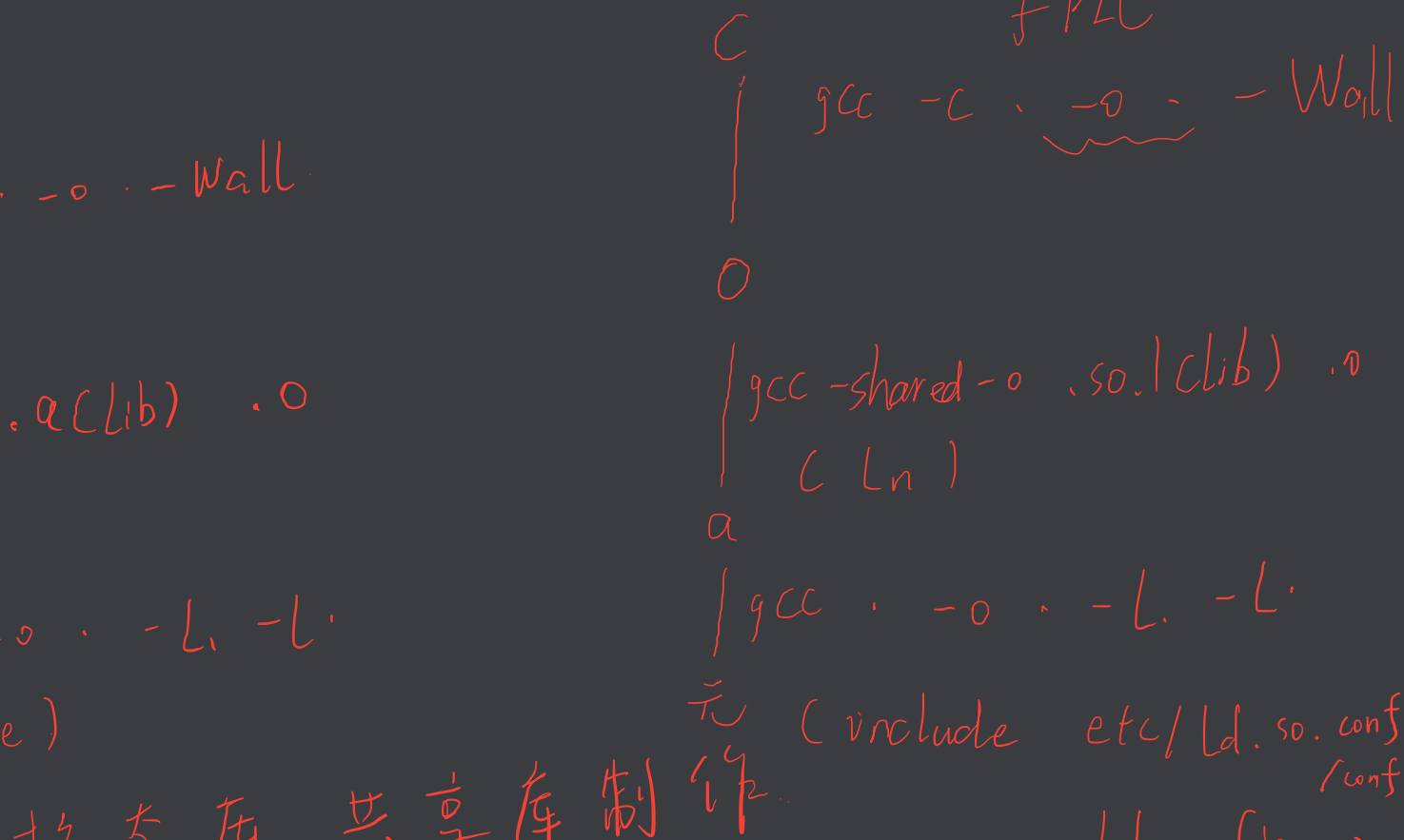
静态库（了解）

静态库创建（熟练）

链接静态库（熟练）

小结




静态库、共享库制作

小

库制作

# 库的概念

库是一个二进制文件，包含的代码可被程序调用

标准C库、数学库、线程库……

库有源码，可下载后编译；也可以直接安装二进制包

/lib　/usr/lib

# 库的知识

库是事先编译好的，可以复用的代码。

在OS上运行的程序基本上都要使用库。使用库可以提高开发效率。

Windows和Linux下库文件的格式不兼容

Linux下包含静态库和共享库

# 静态库特点

编译(链接)时把静态库中相关代码复制到可执行文件中

程序中已包含代码，运行时不再需要静态库

程序运行时无需加载库，运行速度更快

占用更多磁盘和内存空间

静态库升级后，程序需要重新编译链接

# 静态库创建（1）

- 确定库中函数的功能、接口

- 编写库源码hello.c

```
#include  <stdio.h>
void  hello(void) {
   printf( “hello  world\n” );
   return;
}
```

- 编译生成目标文件

```
$ gcc  -c  hello.c  -Wall
```

# 静态库创建（2）

- 创建静态库 hello

```
$ ar  -rsv  libhello.a  hello.o
```

# 静态库创建（3）

ar 参数：

c 禁止在创建库时产生的正常消息

r 如果指定的文件已经存在于库中，则替换它

s 无论 ar 命令是否修改了库内容都强制重新生成库符号表

v 将建立新库的详细的逐个文件的描述写至标准输出

q 将指定的文件添加到库的末尾

t 将库的目录写至标准输出

# 静态库创建（4）

查看库中符号信息

```
$nm  libhello.a

hello.o:
0000000 T  hello
         U  puts
```

nm：显示指定文件中的符号信息

-a 显示所有符号

# 链接静态库

- 编写应用程序test.c

```
#include <stdio.h>
void hello(void);
int main() {
    hello();
    return 0;
}
```

- 编译test.c 并链接静态库libhello.a

```
$ gcc -o test test.c -L. -lhello
$ ./test
 hello world
```

# 小结

库-library

静态库