嵌入式技术

静态库与共享库

邢超

<EC.1>

1 简介

程序生成过程

- 编辑 (源文件)
- 编译 (汇编文件，中间代码文件)

```
gcc -S hello.cpp
```

- 汇编 (目标文件，机器指令码文件)

```
gcc -c hello.cpp
```

- 链接 (可执行文件)

```
gcc hello.cpp
```

<EC.2>

链接

- 静态链接
- 动态链接

<EC.3>

2 静态库/动态库生成

源程序

```
/******  say.c  *******/
#include "stdio.h"
void say()
{
        printf("Say!");
}
/****** test.c  *******/
#include "stdio.h"
void say();
main(){
        say();
}
```

<EC.4>

命令

```
gcc -c say.c
ar -r say.a say.o
gcc test.c say.a -o test
ldd test
gcc -fPIC -shared say.c -o say.so
gcc test.c ./say.so -o test
ldd test
```

<EC.5>

3 动态库使用

动态库位置

- LD_LIBRARY_PATH
- /etc/ld.so.cache
  - ldconfig
  - /etc/ld.so.conf
- 默认路径
  - /lib
  - /usr/lib

<EC.6>

动态加载

```
#include <stdlib.h>
#include <stdio.h>
#include <dlfcn.h>
int main(int argc,char **argv){
  void *handle;      char *error;
  double (*cosine)(double);
  handle = dlopen("/lib/libm.so.6",
     RTLD_LAZY);
  if(!handle){printf("%s\n",dlerror()
     );exit(1);}
  printf("opened␣/lib/libm.so.6\n");
  cosine = dlsym(handle,"cos");
  if((error = dlerror())!=NULL){
    printf("%s\n",error);dlclose(
       handle);
```

```
        printf("closed /lib/libm.so.6\n")
            ;exit(1);}
    printf("%f\n",(*cosine)(2.0));
    dlclose(handle);
    printf("closed /lib/libm.so.6\n");
    return 0;}
/* gcc -o test test.c -ldl
/lib/libm.so.6是动态加载库
/usr/lib/libdl.so是共享库 */
```

<EC.7>

4 思考

思考

- 静态库与共享库/动态库在程序设计中有哪些作用？
- 利用 GNU/Linux 中静态库与共享/动态库设计一个程序。

<EC.8>