

# 7-进程

## 环境表

- 环境表是一个字符指针数组；每个指针包含一个null结束的C字符串的地址；全局变量environ包含该指针数组地址

```
1 extern char **environ
```

## 共享库

- static是静态库；dll是动态库
- 静态库将所有调用的函数拷贝到可执行文件；动态链接库整个系统只有一份副本，装载或运行时装入

```
● lzy@lzy:~/Workspace/APUE/chapter07$ ls -l
total 880
-rwxr-xr-x 1 lzy lzy 16696 Nov  7 11:39 hello_dll
-rwxr-xr-x 1 lzy lzy 871896 Nov  7 11:40 hello_static
-rw-r--r-- 1 lzy lzy     72 Nov  7 11:39 helloc.c
drwxr-xr-x 2 lzy lzy   4096 Nov  7 11:38 hw
```

## setjmp和longjmp

- 跳过若干调用帧，返回到路径中的某个函数中

```
1 #include <setjmp.h>
2 int setjmp(jmp_buf env);
3 void longjmp(jmp_buf env, int val);
```

- jmp\_buf是数组，可以恢复栈状态的所有信息
- 全局变量和静态变量以及volatile变量在longjmp后不变

◦ 开启编译器优化后，自动变量和寄存器变量会使用寄存器优化存储，会发生变化

## getrlimit和setrlimit

- 获取和设置进程资源限制

## ● 更改资源限制

- 任何一个进程可将软限制值更改为小于等于硬限制值
- 任何一个进程可以降低硬限制值，但必须不小于软限制值
- 只有超级用户进程可以提高硬限制值
- RLIM\_INFINITY指定了一个无限量的限制

```
1 #include <sys/resource.h>
2 int getrlimit(int resource, struct rlimit *rlptr);
3 int setrlimit(int resource, const struct rlimit *rlptr);
4
5 struct rlimit {
6     rlim_t rlim_cur;
7     rlim_t rlim_max;
8 };
```

## 习题

- 可执行程序中含有调试的core信息，比原本的数据或文本更大
- 某些UNIX系统实现访问不到数据段的0dan