1-UNIX基础知识

1.1 登录

- 登录信息保存在/etc/passwd文件下
- 登录项：登录名、加密口令、数字用户ID、数字组ID、注释字段、起始目录、shell程序

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
```

1.2 文件和目录

- 所有文件起点是根目录/
- 逻辑上可以认为一个目录项包含一个文件名

```
# 绝对路径
/home/lzy/Desktop
# 相对路径
~/Desktop
```

- 每个进程都有工作目录，也叫当前工作目录（chdir可以改变工作目录）
- 登陆时，工作目录设置为起始目录，起始目录由/etc/passwd中登录项来决定

1.3 输入和输出

- 0 (STDIN_FILENO)，1 (STDOUT_FILENO)，2描述符为标准输入、标准输出和标准错误
- open, read, write, lseek, close提供不带缓冲的IO；都使用文件描述符
- 标准IO函数提供了一个带缓冲的接口：一般命令行缓冲区是行缓冲；文件缓冲区是全缓冲

1.4 程序和进程

- 程序是一个存储在磁盘上某个目录中的可执行文件
- 程序的执行实例被称为进程
- 进程控制函数：fork, exec, waitpid
  - fork是父进程创建子进程，子进程复制父进程的地址空间
  - exec是新进程用自己的地址空间覆盖调用进程的地址空间；**.text, .data, stack, heap**
  - ELF文件加载到内存后初始化堆栈；

```
address:7ffc2b12b5db, env0_ascii_string:SHELL=/bin/bash
address:7ffc2b12b5d3 ~ 7ffc2b12b5da, argument0_ascii_string:./a.out
address:7ffc2b129e50 ~ 7ffc2b12b5d2, padding for align
address:7ffc2b129e49 ~ 7ffc2b129e4f, String identifying platform:x86_64
address:7ffc2b129e39 ~ 7ffc2b129e48, rand_bytes[16]: 56 159 54 171 40 53 16
151 19 184 231 202 222 117 253 173
padding
address:7ffc2b129cf8 ~ 7ffc2b129e28, auxivilary vector
address:7ffc2b129bd8 ~ 7ffc2b129cf0, environment vector
address:7ffc2b129bc8 ~ 7ffc2b129bd0, argv
address:7ffc2b129bc0, argc
```

- **auxiliary vector** 是内核ELF二进制加载器提供给用户空间的一些信息的集合，包括可执行的入口地址、线程的gid、线程uid、vdso入口地址等等。普通应用程序通常不需要关心这些信息，目前主要是动态链接器（ld-linux.so）在使用这些信息

1.5 出错处理

- 之前定义为extern int errno；多个线程共享地址空间时很难判断是哪个线程出现错误
- 多线程存取的errno

```
extern int *__errno_location(void);
#define errno (*__errno_location());
```

- 将errno变为TLS，每个线程都有自己的errno

```
/* errno-log.c */
#include <errno.h>
#include <tls.h>

int *
__errno_location (void)
{
  return &errno;
}
libc_hidden_def (__errno_location)

/* errno.h */
extern __thread int errno attribute_tls_model_ie;
```

- 错误处理函数perror，strerror

```
#include <string.h>
char* strerror(int errnum);

#include <stdio.h>
void perror(const char* msg);
```

1.6 用户标识

- 用户ID、组ID（组文件将组名映射为组ID，/etc/group）、附属组ID

1.7 信号

进程处理信号方式

- 忽略信号：由于某些信号由硬件产生，忽略该信号导致后果不确定
- 按系统默认方式处理：对于除0操作，系统默认终止进程
- 提供函数，信号发生时调用该函数。

1.8 时间值

- 日历时间 (time_t)
- 进程时间 (clock_t)
  - 时钟时间：进程运行的时间总量
  - 用户CPU时间
  - 系统CPU时间

1.9 系统调用

- 系统调用常常提供一种最小的接口；库函数通常提供比较复杂的功能（如sbrk系统调用和malloc库函数）