



2-UNIX标准

2.1 运行时限制

- `_SC_`开始的常量值作为标识运行时限制的`sysconf`参数
- `_PC_`开始的常量用作标识运行时限制的`pathconf`和`fpathconf`参数
- 如果`name`不是合适的常量，3个函数返回-1，`errno`置为`EINVAL`
- 如果提示该值不确定，返回-1，但是不设置`errno`

```
1 #include <unistd.h>
2 long sysconf(int name);
3 long pathconf(const char *pathname, int name);
4 long fpathconf(int fd, int name);
```

2.2 不确定的运行时限制

- 某些限制值可能不确定；编译时可能不能使用，运行时也可能是未定义的

2.3 选项

- 为了可移植性，对限制的处理，POSIX.1定义了3种处理选项的方法
 - 编译时定义在`<unistd.h>`
 - 与文件或目录无关的运行时选项用`sysconf`

- 与文件或目录有关的运行时选项用pathconf或fpathconf
- 对每个选项有3种可能平台支持状态
 - 符号常量无定义或者值为-1，该平台编译时不支持相应选项
 - 符号常量大于0，该平台支持该选项
 - 符号定义值为0，必须调用sysconf, pathconf, fpathconf来判断相应选项是否受到了支持

2.4 功能测试宏

- _POSIX_C_SOURCE以及_XOPEN_SOURCE被称为功能测试宏
- 排除任何实现专有的定义

2.5 基本数据类型

类型	说明
clock_t	进程时间的时钟滴答计数器
comp_t	压缩的时钟滴答
dev_t	设备号（主、次）
fd_set	文件描述符集
fpos_t	文件位置
gid_t	数值组ID
ino_t	i节点编号
mode_t	文件类型
nlink_t	目录项的链接计数
off_t	文件长度和偏移量
pid_t	进程ID和进程组ID
pthread_t	线程ID
ptrdiff_t	两个指针相减的结果
rlim_t	资源限制
sig_atomic_t	可原子访问的数据类型

sigset_t	信号集
size_t	对象长度
ssize_t	返回字节计数的函数
time_t	日历时间的计数器
uid_t	数值用户ID
wchar_t	能表示所有不同的字节码

2.6 补充

- 如果某个限制十分巨大或者不确定的话，使用getrlimit来获取进程的限制

```

1  long open_max(void) {
2      struct rlimit rl;
3      if (openmax == 0) {
4          errno = 0;
5          /* LONG_MAX过大 */
6          if ((openmax = sysconf(_SC_OPEN_MAX)) < 0 || openmax == LONG_MAX) {
7              /* 获取进程的最大限制 */
8              if ((openmax = getrlimit(RLIMIT_NOFILE, &rl)) < 0)
9                  err_sys("can not get file limit\n");
10             else if (openmax == RLIM_INFINITY)
11                 openmax = OPEN_MAX_GUESS;
12             else
13                 openmax = rl.rlim_max;
14         }
15     }
16     return (openmax);
17 }
```