
信号扩展

- signal函数只能提供简单的信号安装操作。使用signal函数处理信号比较简单，只要把要处理的信号和处理函数列出即可。
- signal函数主要用于前面32种不可靠(在Linux系统中不可靠的信号是指信号可能会丢失)、非实时信号的处理，并且不支持信号传递信息。
- Linux提供了功能更强大的sigaction函数，此函数可以用来检查和更改信号处理操作，可以支持可靠、实时信号的处理，并且支持信号传递信息。

➤ 可靠、不可靠信号

从UNIX系统继承过来的信号（SIGHUP ~ SIGSYS）都是不可靠信号，不支持排队（多次发送相同的信号，进程可能只能收到一次，可能会丢失）。SIGRTMIN至SIGRTMAX的信号支持排队（发多少次，就可以收到多少次，不会丢失），故称为可靠信号。

➤ 实时、非实时信号

可靠信号就是实时信号，非可靠信号就是非实时信号。

- 实时信号（可靠信号）的优势
 - 多个实时信号可以同时发生并传送给相应的进程。
 - 实时信号可以传参(整数或任意类型的地址)。
 - 实时信号可以保证信号传递的顺序(传递顺序和信号产生的顺序相同)。

➤ #include <signal.h>

```
int sigqueue(pid_t pid, int sig,  
              const union sigval value);
```

功能:

给指定进程发送信号。

参数:

pid: 进程号。

sig: 信号的编号。

value: 通过信号传递的参数。

➤ 信号传递的参数:

```
union sigval  
{  
    int    sival_int;  
    void *sival_ptr;  
};
```

返回值:

成功返回 0, 失败返回 -1

➤ #include <signal.h>

```
int sigaction(int signum,  
              const struct sigaction *act,  
              struct sigaction *oldact);
```

功能:

检查或修改指定信号的设置 (或同时执行这两种操作)。

参数:

signum: 信号编号。

act: 新的信号设置指针。

oldact: 旧的信号设置指针。

➤ sigaction函数

➤ 如果act指针非空，则要改变指定信号的设置，如果oldact指针非空，则系统将此前指定信号的设置存入oldact。

➤ 返回值：

成功：0

失败：-1

信号设置结构体:

```
struct sigaction
```

```
{
```

```
    /*旧的信号处理函数指针*/
```

```
    void (*sa_handler)(int signum) ;
```

```
    /*新的信号处理函数指针*/
```

```
    void (*sa_sigaction)(
```

```
        int signum, siginfo_t *info, void *context);
```

```
    sigset_t sa_mask; /*信号阻塞集*/
```

```
    int sa_flags; /*信号处理的方式*/
```

```
};
```

➤ sa_handler、sa_sigaction:

信号处理函数指针:

➤忽略该信号: SIG_IGN

➤执行系统默认动作: SIG_DFL

➤自定义信号处理函数: 处理函数名

注: 应根据情况给sa_sigaction、sa_handler两者之一赋值。

sa_mask:

信号处理函数执行期间要阻塞的信号集(但不阻塞SIGKILL、SIGSTOP)。

➤ `sa_flags`: 用来决定信号的行为。

➤ 不指定 `SA_SIGINFO`:

信号的处理函数应由 `sa_handler` 设置。可完成与 `signal` 函数一样的功能。

➤ 指定 `SA_SIGINFO`:

信号的处理函数应由 `sa_sigaction` 设置。它的功能更强大，除了获取基本的信号值外，还可以获取此信号发送者及接收者的部分信息。

例: [07_sigaction_1.c](#) [07_sigaction_2.c](#)

➤ 信号处理函数:

```
void (*sa_sigaction)(int signum,  
                     siginfo_t *info, void *context);
```

参数:

- signum: 信号的编号。
- info: 记录信号发送进程信息的结构体。
- context: 可以赋给指向ucontext_t类型的一个对象的指针，以引用在传递信号时被中断的接收进程或线程的上下文。

进程信息结构体路径:

/usr/include/i386-linux-gnu/bits/siginfo.h

发送进程信息的结构体: siginfo_t

接收进程或线程的上下文结构体: ucontext_t

例: 07_sigaction_3_rcv.c
07_sigaction_3_snd.c