


第5章 信号

——早期信号处理函数signal

主 讲：黄 茹

信号处理方式

- 按系统默认方式处理
- 忽略信号
- 捕捉信号

选择信号处理方式

signal	
功能	简单的信号处理
头文件	/usr/include/signal.h
函数原型	sighandler_t signal(int signum, sighandler_t handler);
参数	signum　要响应的信号
	handler　信号发生时要执行的操作
返回值	非-1　执行成功，返回以前的信号处理函数指针
	-1　　遇到错误

```
typedef void (*sighandler_t)(int);
```

三个常用的伪函数

```
/* Fake signal functions.  */
#define SIG_ERR ((_ _sighandler_t) -1)        /* Error return.  */
#define SIG_DFL ((_ _sighandler_t) 0)         /* Default action.  */
#define SIG_IGN ((_ _sighandler_t) 1)         /* Ignore signal.  */
```

示例程序——忽略信号

```
#include <stdio.h>
#include <signal.h>

main( ) {
    int i=10;
    signal(SIGINT,SIG_IGN);
    printf("waiting for signal...\n");
    while(i>0)  {
        sleep(1);      i--;
        printf("Now you can't stop this program by Ctrl+c!\n");
    }
}
```

示例程序——捕捉信号

```
#include <stdio.h>
#include <signal.h>

void capture(int signum) {
    printf("SIGINT is captured!\n");
}

main( ) {
    int i=10;
    signal(SIGINT,capture);
    printf("waiting for signal...\n");
    while(i>0)    {
          sleep(1);          i--;
    }
}
```

示例程序——恢复默认处理


```
#include <stdio.h>
#include <signal.h>

main(){
   int i=5;
   signal(SIGINT,capthendfl);
   printf("waiting for signal...\n");
   while(i>0)   {   sleep(1);     i--;   }
   i=5;
   while(i>0)   {   sleep(1);     i--;   }     }
```

```
void capthendfl(int signum){
   printf("SIGINT is captured!\n");
   signal(SIGINT, SIG_DFL);
   printf("SIGINT now is defaulted!\n");
}
```

signal函数的问题

- 各个版本的实现不同，有可能出现的问题：

1. 处理函数每次使用之后都会被禁用么？

   如果signal函数的版本是报警器方式的，那么信号处理函数格式应设置为：

```
void handler(int signo)  {
   signal(SIGINT,handler)
    ……          //信号处理的内容
}
```

2. 只获取当前信号处理方式时，操作复杂
3. 进程在处理一个SIGX时，第二个SIGX到达会发生什么？处理SIGX时有信号SIGY到达，系统如何处理?
4. 如果消息到来时，进程正在处理getchar之类的低速系统调用，会发生什么?
5. 其他问题

解决办法：使用sigaction系统调用

谢谢大家！