

# Упражнение 7

## Прихващане на сигнал

```
#include <signal.h>
typedef void (*sighandler_t)(int) - указател към функция с параметър int и
резултат void
sighandler_t signal(int signum, sighandler_t handler) - задава реакция при
сигнал

int sigaction(int signum, const struct sigaction *act, struct
sigaction *oldact) - задава и/или получава реакция при сигнал

struct sigaction {
    void      (*sa_handler)(int); - реакция при сигнал
    void      (*sa_sigaction)(int, siginfo_t *, void *); - реакция при сигнал
    sigset_t   sa_mask; - маска на сигнали, които се блокират
    int        sa_flags; - флагове (SA_SIGINFO)
    void      (*sa_restorer)(void); - не се ползва
}
```

SIG\_IGN - игнориране

SIG\_DFL - реакция по подразбиране

## Сигнали

SIGHUP 1 Term - прекъсване на връзката с управляващия терминал  
 SIGINT 2 Term - "прекъсване" от клавиатурата (Ctrl+C)  
 SIGQUIT 3 Core - "излизане" от клавиатурата (Ctrl+\)  
 SIGILL 4 Core - недопустима инструкция  
 SIGABRT 6 Core - сигнал от abort()  
 SIGFPE 8 Core - препълване при операция с плаваща точка  
 SIGKILL 9 Term - сигнал за убиване  
 SIGSEGV 11 Core - недопустима операция с паметта  
 SIGPIPE 13 Term - писане в канал без читатели  
 SIGALRM 14 Term - сигнал от alarm()  
 SIGTERM 15 Term - сигнал за прекратяване  
 SIGUSR1 10 Term - потребителски сигнал 1  
 SIGUSR2 12 Term - потребителски сигнал 2  
 SIGCHLD 17 Ign - завършване на син  
 SIGCONT 18 Cont - продължаване ако е спрял  
 SIGSTOP 19 Stop - спиране на процес

## Изпращане на сигнал

```
#include <sys/types.h>
#include <signal.h>
int kill(pid_t pid, int sig) - изпраща сигнал
```

```
#include <unistd.h>
int pause(void) - чака сигнал
```

```
#include <unistd.h>
unsigned int alarm(unsigned int seconds) - планира изпращането на SIGALRM
```