

## Б9. Сигнали: <signal.h>

Заглавният файл <signal.h> предоставя начини за управление на условия, възникнали по време на изпълнението на програмата (например прекъсващ сигнал от външен изходен код или грешка при самото изпълнение).

```
void (*signal(int sig, void (*handler) (int))) (int)
```

signal определя как ще се обработват следващите я сигнали. Ако handler има стойност SIG\_DFL, ще се използва поведението по подразбиране за съответната реализация; ако стойността му е SIG\_IGN, сигналът ще бъде пренебрегнат. В противен случай се извиква функцията, към която сочи handler; като аргумент на функцията се подава типът на сигнала. Валидните сигнали са:

SIGABRT	необичайно прекратяване на програмата, например от abort
SIGFPE	аритметична грешка, например деление на нула или надвишаване на зададени граници
SIGILL	грешка, свързана с функция, например грешна инструкция
SIGINT	интерактивно внимание, например прекъсване
SIGSEGV	грешка при заделяне на паметта, например опит за получаване на достъп извън границите на паметта
SIGTERM	подадено към програмата потвърждение за прекратяване на изпълнението

signal връща предишната стойност на handler за определен сигнал. Ако възникне грешка, signal връща SIG\_ERR.

Когато един сигнал sig се появява постоянно, сигналът възстановява поведението си по подразбиране; в такъв случай се извиква функцията, която управлява сигнала, все едно е използван записът (\*handler) (sig). Ако функцията върне стойност, изпълнението продължава от мястото, където се е появил сигналът.

Първоначалната стойност на сигналите се определя от реализацията.

```
int raise(int sig)
```

raise изпраща сигнала sig към програмата; ако изпращането на сигнала не бъде успешно, raise връща ненулева стойност.