

Б8. Нелокални преходи: <setjmp.h>

Декларациите в <setjmp.h> предоставят начин да се избегне обикновеното извикване на една функция. Те връщат някаква последователност, която обикновено разрешава моменталното връщане в началото от някоя функция, вградена надълбоко в дадена конструкция.

```
int setjmp(jmp_buf env)
```

Макросът `set jmp` запазва в `env` информация за състоянието, която впоследствие се използва от `longjmp`. При директно извикване от `set jmp`, върнатата стойност е нула; при следващо извикване от `longjmp`, върнатата стойност е различна от нула. Извикването на `setjmp` се извършва в определен контекст, обикновено в тестовата част на `if`, `switch` или на цикъл; освен това макросът се използва само в елементарни релационни изрази.

```
if (setjmp (env) == 0)
```

```
/* минава през тази част на if при директно извикване */
```

```
else
```

```
/* минава през тази част на if при извикване на longjmp */
```

```
void longjmp(jmp_buf env, int val)
```

`longjmp` възстановява състоянието, записано от последното извикване на `set jmp`, като за тази цел използва информацията, съхранена в `env`. Изпълнението на програмата продължава, сякаш функцията `setjmp` е била току-що изпълнена и върнатата от нея стойност е различна от нула (`val`). Задължително е функцията, в която се намира `setjmp`, да не е приключила изпълнението си. Обектите, до които има достъп, притежават стойностите, които са имали при извикването на `longjmp`. Единственото изключение е, че автоматичните променливи във функцията, извикваща `setjmp`, които не са от тип `volatile`, стават неопределени, ако са били променени след извикването на `set jmp`.