

**NAME**

handlecond() - sets a function to handle a condition

**SYNOPSIS**

```
#include <schily/sigblk.h>
```

**void**

```
handlecond(signame, sp, func, arg1)
```

```
char *signame;
```

```
SIGBLK *sp;
```

```
int (*func)();
```

```
long arg1;
```

**void**

```
starthandlecond(sigfirst)
```

```
SIGBLK *sigfirst;
```

**void**

```
unhandlecond(sigfirst)
```

```
SIGBLK *sigfirst;
```

**DESCRIPTION**

**handlecond()** sets up a handler for a condition. The user must explicitly allocate the condition block and pass it to the function.

**handlecond()** establishes function `func` as the condition handler for the condition `signame`. `arg1` is passed to `func` at the time of condition signalling. `signame "any_other"`, catches all conditions.

The specified function is called with the actual condition being signalled, with arguments from the latest **handlecond()** call for that function in the given frame and from the call to **raisecond()**. For instance:

```
int func (signame, arg1, arg2)
```

```
    char *signame; /* the actual condition being
```

```
                   /* signalled */
```

```
    int arg1;      /* arg1 comes from the handle
```

```
                   /* call that set up the handler */
```

```
    int arg2;      /* arg2 comes from the raise call */
```

If the function returns TRUE (non-zero), it is assumed that the condition has been successfully

handled; otherwise, the condition is signalled farther down the stack.

**starthandlecond()** sets an initial marker to the current stack frame. It needs to be called before **handlecond()** may be used and the argument needs to be a variable local to the current function.

**unhandlecond()** needs to be called before a function may call **return** or fall out of the function body. The argument needs to be the same as used for the **starthandlecond()** function.

## RETURNS

none

## NOTES

Be careful when declaring args to func if they are not long; both args will occupy sizeof(long) bytes in the arglist. If FALSE is returned by a condition handler, and there is an any\_other handler in the same block, the any\_other handler will be invoked (only once, since it too may return FALSE).

To revert a condition handler simply use NULL as func.

SIGBLK, defined in <schily/sigblk.h>, must be included. **handlecond()** is frequently used with **longjmp()** and **setjmp()**.

Eah function which calls **handlecond()** must call **unhandlecond()** before it returns. Otherwise the return will fail or cause a core dump.

## BUGS

**handlecond()** makes the framepointer odd, to mark the current stack frame.

This confuses programs like **adb(1)** and **dbx(1)** because the debuggers scan the stack frame to get the call stack.