

NAME           intr -- set interrupt handling

SYNOPSIS       sys intr; arg / intr = 27.

intr(label)

int \*label;

DESCRIPTION   When arg is 0, interrupts (ASCII DELETE) are ignored. When arg is 1, interrupts cause their normal result, that is, force an exit. When arg is a location within the program, control is transferred to that location when an interrupt occurs.

After an interrupt is caught, it is possible to resume execution by means of an rti instruction; however, great care must be exercised, since all I/O is terminated abruptly upon an interrupt. In particular, reads of the typewriter tend to return with 0 characters read, thus simulating an end of file.

From C, the interface is slightly different. An argument of 0 or 1 has the afore-mentioned effect of suppressing and forcing termination on interrupts respectively. An argument which is a label causes control to be sent to the label when an interrupt occurs; however the stack is set to have the same value as it did when intr was called; thus control returns to the complete execution environment pertaining at the time of the call to intr. It is an error to call intr in a subroutine and return from that subroutine, since if an interrupt occurs control will be returned to a non-existent environment.

It is not possible (with the C version of this call) to resume execution after an interrupt.

SEE ALSO       quit(II)

DIAGNOSTICS   --

BUGS           --