NAME
    splhi, spllo, splx, islo - enable and disable interrupts

SYNOPSIS
    int spllo(void)

    int splhi(void)

    void splx(int x)

    int islo(void)

DESCRIPTION
    These primitives enable and disable maskable interrupts on the current processor. Generally, device drivers should use ilock (see lock(10.2)), sleep(10.2), or the functions in qio(10.2) to control interaction between processes and interrupt handlers. Those routines (but not these) provide correct synchronisation on multiprocessors.

    Spllo enables interrupts and returns a flag representing the previous interrupt enable state. It must not normally be called from interrupt level.

    Splhi disables all maskable interrupts and returns the previous interrupt enable state. The period during which interrupts are disabled had best be short, or real-time applications will suffer.

    Splx restores the interrupt enable state state to x, which must be a value returned by a previous call to splhi or spllo.

    Islo returns true (non-zero) if interrupts are currently enabled, and 0 otherwise.

SEE ALSO
    lock(10.2), qio(10.2), sleep(10.2), intrenable(10.2)