The signal.c file contains a portable implementation of the signal() function, which is used to change the action taken by a process when it receives a specific signal.

The file contains a single function, Signal, which takes two parameters: signo, the signal number, and func, a pointer to the signal handling function. This function is a wrapper around the sigaction() system call, which is used to change the action taken by a process on receipt of a specific signal.

The Signal function first initializes a sigaction structure act with the desired signal handling function func, and an empty signal set. It then sets the SA_RESTART flag if the signal is not SIGALRM, to automatically restart system calls on a signal. For SIGALRM, it sets the SA_INTERRUPT flag on systems where it is available, to interrupt system calls on a signal.

The function then calls sigaction() to change the action for the signal signo to the act structure, and saves the old action in the oact structure. If sigaction() returns an error, the function returns SIG_ERR. Otherwise, it returns the old signal handler.