Sam Fellers <sfellers>
Lab6

3. To make the async callback registration function, I first added a field to the process table that stores the function pointer to register our callback.

Next I created regcallback.c and the system call regcallback() with the parameter of a function pointer. The regcallback function registers the function pointer in that processes entry of the process table. Inside resched() we call the callback function if the process has received a message.

4. For the different types of signalling I added a few fields to the process table, prcbtime which keeps track of wall time since A callback was registered of type SIGXTIME. Prcbflag, which stores the type of signal, and prdeadbaby which stores a PID when a child process has been killed. Whether the callbacks are run is decided based on the logic in resched() after they are called the flags are reset.

I implemented the waitchld() function by making a function very similiar to receive(), however instead it checks to see if a process is in the WAITSIG state, if it still is then we resched(). Inside kill we look for processes that are dying whose parent called waitchld(), we ready them. And the waitchld() returns the pid of that child.