

LEARNING DIARY&USER GUIDE__ASSIGNMENT1.VERSION2 < exec() code replacement version>

/* By Yunfeng He; 2012.11.13; UnixC Assignment 1_version2 */

INSTALLATION:

- In the folder exec, there's a Makefile. Make to generate two executables, with "morsecoder" the main program. exec() would swap code for the child process and the child process would execute "decoder".
- Make clean to clean executables and the copied file, together with the log file named "LOGFILE".

USAGE:

- The program "morsecoder" together with "decoder", do exactly the same job with my "morsetranscoder" program.
- Type in ./morsecoder [filename] to run the program.
- For example ./morsecoder morsecoder.c would make a copy of morsecoder.c and the copied file is named as morsecoder.cCOPY; program would also append log messages to the file LOGFILE.
- Log messages are written in order of time. However to make the log looks simpler, no logging time is kept. LOGFILE simply provide a hint on what the program has been doing. Those log messages were used as debugging messages.

DESIGN AND IMPLEMENTAION:

The morsetranscoder version 1 works briefly like this:

- The main process set signal handlers for both processes(before fork)
- After fork, the parent call the senderprocess() to open file and send morsecode sequence as it read characters; The child calls the decoderprocess() to parse the morsecode sequence and giving 'Go' signal back to the parent at the same time.(flow control)
- After setting up handlers, basically, each process is doing its own job based on independent resources. This makes the exec() code replacement very simple to implement, without changing the entire program too much.

The morsetranscoder version 2 have following things changed:

- Instead of calling decoderprocess() directly in main.c(like version 1 did), the child process would use exec() to replace the code with the executable "decoder".
- Signal handlers for the decoder process is loaded in the decoder program rather than in the main program. The source files decoder.c and decoder.h will generate executable file "decoder".
- The name of the file to be copied is passed to the executable "decoder" as argument. the argument array is loaded in morsecoder.c, to be exact morsecoder, before exec() is called.
- If the "decoder" is not found by exec(), then the child exit before send SIGKILL to terminate the parent process. This is because the parent process(the sender) is waiting for 'Go' from the child. If the decoder is

not properly loaded and executed, the parent would just hang there forever.