Write your own shell

Homework objectives:

- Learn how to use system calls
- Learn how to create child processes
- Learn how to catch signals
- Learn basic I/O redirection

Write your own shell

Homework objectives:

- Learn how to use system calls
- Learn how to create child processes
- Learn how to catch signals
- Learn basic I/O redirection

The programming part

For this assignment you are to write a simple program that will act as a shell. The program shall

- Display a command prompt and read in a command line from the user. You should use "361› " as your prompt.
- Parse the command line into arguments, creating an array of character pointers, where array[0] points to the actual command and rest of the array elements point to the arguments to the command (Similar to main()'s argv[])
- Fork off a child and have the child load the requested program by passing the argument vector created in step 2 to exec() family of system calls. The parent should report the PID of the child before proceeding to the next step. You should use the format "PID: 5", where the child's PID is 5.
- Wait for the child to complete executing and report that it ended and its exit value if available, by printing "Exit: 0" (assuming its exit value is 0).
- Repeat for first step forever till user enters the command `exit`
- Your shell should also support basic I/O redirection line the unix shell.
    - `command › filename` Redirects the output of command to filename. The existing contents of filename are overwritten.

    - `command >> filename`Redirects the output of command to filename. The output from command is appended to contents of filename. Existing contents are not overwritten.
    - `command < filename` Command reads its input from filename instead of from stdin.
- Your shell should handle the following signals:
- SIGINT - Generated by Ctrl-C. This signal allows a user to teminate a running program. Your shell should not exit when user presses Ctrl-C but simply report that SIGINT signal has been received by the shell by printing "SIGINT handled".
- SIGTSTP - Generated by Ctrl-Z. Your shell should not exit when user presses Ctrl-Z but simply report that SIGTSTP signal has been received by the shell by printing "SIGTSTP handled".