```c
/* CS 352 -- Mini Shell!
 *
 *   Matt Forbes - Assignment 1 - 9/23/11
 *
 */

#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <errno.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/wait.h>

#include "proto.h"

/* Constants */

#define LINELEN 1024

/* Prototypes */

void processline (char *line);

/* Shell main */

int main (void)
{
    char   buffer [LINELEN];
    int    len;

    while (1) {

        /* prompt and get line */
        fprintf (stderr, "%% ");
        if (fgets (buffer, LINELEN, stdin) != buffer)
            break;

        /* Get rid of \n at end of buffer. */
        len = strlen(buffer);
        if (buffer[len-1] == '\n')
            buffer[len-1] = 0;

        /* Run it ... */
        processline (buffer);

    }

    if (!feof(stdin))
        perror ("read");

    return 0;           /* Also known as exit (0); */
}


void processline (char *line)
{
    pid_t  cpid;
    int    status,
           argc;
    char   **argv;

    argc = arg_parse(line, &argv);

    /* when no arguments, do nothing */
    if (argc == 0)
        return;
```

```
    /* try calling a builtin, return if successful */
    if (try_builtin(argc, argv))
        return;

    /* Start a new process to do the job. */
    cpid = fork();
    if (cpid < 0) {
        perror ("fork");
        return;
    }

    /* Check for who we are! */
    if (cpid == 0) {
        /* We are the child! */
        execvp(argv[0], argv);
        perror ("exec");
        exit (127);
    }

    /* free argv when parent */
    free(argv);

    /* Have the parent wait for child to complete */
    if (wait (&status) < 0)
        perror ("wait");
}
```