```
/*****************************************************************************
/
/      filename:  myshell.c
/
/   description:  Impliments the shell for Project 1.
/
/        author:  Paladino, Zac
/      login id:  cps346−n1.16
/
/         class:  CPS 346
/    instructor:  Perugini
/    assignment:  PJ #1
/
/      assigned:  January 28, 2009
/           due:  February 20, 2009
/
*****************************************************************************/

#include<stdio.h>
#include<stdlib.h>
#include<limits.h>
#include <unistd.h>
#include <fcntl.h>
#include <string.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <sys/stat.h>

#include "builtin.h"
#include "utils.h"
#include "makeargv.h"
#define DELIMITERS " \t"

typedef int bool;
#define TRUE 1
#define FALSE 0
#define CREATE_FLAGS (O_WRONLY | O_CREAT | O_TRUNC)
#define CREATE_READ_FLAG (O_RDONLY)
#define CREATE_MODE (S_IRUSR | S_IWUSR | S_IRGRP | S_IROTH)

int main(int argc, char** argv){
   FILE* input;
   if(argc == 1){
     input = stdin;
   }
   else if(argc == 2){
     input = fopen(argv[1],"r");
   }
   else{
     perror("Usage");
     exit(1);
   }
   int numtok=0, numtokens=0, lsize=0;
   char p;
   char** newargv;
   static bool REDSO = FALSE, REDSI = FALSE, EXIT = FALSE;

   char* line = malloc(MAX_CANON);
   char** toklin;
   char* newl = NULL;
   char* opf = NULL;
   char* ipf = NULL;
   int i = 0;

   static int fd;
   static int fdi;
   static int fdr;

   void* x = NULL;
```



```
   fprintf(stderr, "Zshell $: ");
   x = fgets (line, MAX_CANON, input);
   if (x) {
      if((strlen(line)−1)!=0){
        line[strlen(line)−1] = '\0';
      }
   }
   while((x != NULL)&&((strcmp(line, "quit")) != 0)) {
       newl = malloc(MAX_CANON);
       opf = malloc(MAX_CANON);
       ipf = malloc(MAX_CANON);
       stripIO(&line,&toklin,&newl,&opf,&ipf,&numtok);
       if((numtokens = makeargv(newl, DELIMITERS, &newargv))==−1){
          perror("makeargv error");
          return 1;
       }
       RedIO(&toklin,&opf,&ipf,&REDSI,&REDSO,&numtok,&fd,&fdi,&fdr,&EXIT);
       if((strstr(newargv[0],"cd"))){
         if(newargv[1]){
           cd(&newargv);
         }
         else{
           perror("Need to Include a directory");
         }
         EXIT=TRUE;
       }
       else if((strstr(newargv[0],"environ"))){
         env();
         EXIT=TRUE;
       }
       else if((strstr(newargv[0],"echo"))){
         Echo(&newargv,&numtokens);
         EXIT=TRUE;
       }
       else if((strstr(newargv[0],"pause"))){
         Pause();
         EXIT=TRUE;
       }
       else if(!EXIT && !(strstr(newargv[0],"\n"))){
        execproc(&newargv,&line,&REDSO,&REDSI,&fdr,&x,&input);
       }
       if(EXIT && (strcmp(newargv[0],"\n")!=−1)){
         EXIT = FALSE;
         fprintf(stderr, "Zshell $: ");
         if(REDSO){
            freopen("/dev/tty", "w", input);
            REDSO = FALSE;
          }
         if(REDSI){
            dup2(fdr, STDIN_FILENO);
            close(fdr);
            REDSI = FALSE;
         }
         x = fgets (line, MAX_CANON, input);
         if (x) {
           if((strlen(line)−1)!=0){
             line[strlen(line)−1] = '\0';
           }
         }
       }
   }
   return 0;
}
```