```
Cameron Wallace
CS 347 Spring 2020
Assignment 1
```

```
/* Cameron Wallace
 * CS 352 -- Micro Shell!
 *
 *   Sept 21, 2000,  Phil Nelson
 *   Modified April 8, 2001
 *   Modified January 6, 2003
 *   Modified January 8, 2017
 *
 */

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

/* Constants */

#define LINELEN 1024

/* Prototypes */

void processline(char *line);
char **arg_parse(char *line, int *argcptr);
char *removeQuotes(char *line, int n, int quotes, char **argarr);

/* 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;

  int argc;
  char **args = arg_parse(line, &argc);

  /* Start a new process to do the job. */
  cpid = fork();
  if (cpid < 0)
  {
    /* Fork wasn't successful */
    //free(args); breaks
    perror("fork");
    return;
  }

  /* Check for who we are! */
  if (cpid == 0)
  {
    /* We are the child! */
    //execlp (line, line, (char *)0);
    execvp(*args, args);
    /* execlp reurned, wasn't successful */
    perror("exec");
    fclose(stdin); // avoid a linux stdio bug
    exit(127);
  }

  /* Have the parent wait for child to complete */
  if (wait(&status) < 0)
  {
    free(args);
    /* Wait wasn't successful */
    perror("wait");
  }
}

char **arg_parse(char *line, int *argcptr)
{
  /* line points to character array containing command to be processed */
  /* argcptr points to int variable representing number of args in line */
  /* return value: pointer to malloced area that points into line parameter  */
  /* call malloc(3) only once */

  int argc = 0;
  int i = 0;
  int inquote = 0;
  int quotec = 0;

  // Count args & quotes
  while (line[i] != 0)
  {
    while (line[i] == ' ') // skip spaces
      i++;
    if (line[i] != ' ' && line[i] != 0) // start arg
    {
      argc++;
      while (line[i] != ' ' && inquote == 0) // find end of arg
```

```
      {
        if (line[i] == 0)
          break;
        if (line[i] == '\"') // find start quote
        {
          quotec++;
          inquote = 1;
          i++;
          while (inquote == 1)
          {
            if (line[i] == '\"') // found end quote
            {
              quotec++;
              inquote = 0;
            }
            i++;
          }
        }
        if (line[i] == ' ')
          break;
        i++;
      }
    }
  }
  if (argc == 0)
    return NULL;
  if (argc == 0 || quotec % 2 == 1)
  {
    return NULL;
  }
  // Allocate memory
  char **argarr = (char **)malloc((argc + 1) * sizeof(char *));

  i = 0; // src
  int dest = 0; // dest
  int ac = 0;

  const int len = strlen(line);

  while (line[i] == ' ') i++; // skip lead spaces
  // Assign pointers and 0s
  while (line[i] != 0 && i < len)
  {
    if (line[i] != ' ')  // start arg
    {
      argarr[ac++] = &line[dest]; // assign pointer to start of arg
      while (line[i] != ' ') // loop until we hit a space
      {
        if (line[i] == '\"') // find start quote
        {
          i++; // get i off quote
          while (line[i] != 0 && line[i] != '\"') // loop until end or terminati
ng quote
          {
            if (line[i] != '\"') // replace characters that aren't "
            {
              line[dest++] = line[i];
            }
            i++;
```

```
          }
          line[dest] = line[i++];
          continue;
        }
        line[dest++] = line[i++];
      }
      line[dest++] = 0; // assign pointer to end of arg
    }
    i++; // i is a space, increment
  }
  argarr[ac] = NULL;

  *argcptr = argc;
  return argarr;
}
```

```
Script started on Fri Apr 10 23:19:09 2020

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
^[[?1034h[Cameron]@23:19:09 $ exit^H^H./a.out^Hgcc -g -Wall ush.c
[Cameron]@23:19:11 $ gcc -g -Wall ush.c^H^Hexit^[[K^H^H./a.out
%  echo hello world
hello world
% ls -l
total 136
-rwxr-xr-x  1 Cameron  staff  13760 Apr 10 23:19 a.out
drwxr-xr-x@ 3 Cameron  staff     96 Apr  9 16:12 a.out.dSYM
-rw-r--r--@ 1 Cameron  staff   1022 Apr  9 11:37 manNotes.rtf
-rw-r--r--@ 1 Cameron  staff   3345 Apr 10 17:15 microshell.c
-rw-r--r--@ 1 Cameron  staff   1540 Apr  9 12:03 microshell.c~
-rw-r--r--  1 Cameron  staff      0 Apr 10 20:38 out.txt
-rw-r--r--  1 Cameron  staff     43 Apr 10 23:19 typescript
-rw-r--r--  1 Cameron  staff  24081 Apr 10 20:44 typescript.pdf
-rw-r--r--@ 1 Cameron  staff   3176 Apr  9 21:50 unixHelpNotes.rtf
-rw-r--r--@ 1 Cameron  staff   3605 Apr 10 23:18 ush.c
-rw-r--r--@ 1 Cameron  staff   3495 Apr 10 15:37 ush.c~
%  ls
a.out                   microshell.c            typescript              ush.c
a.out.dSYM              microshell.c~           typescript.pdf          ush.c~
manNotes.rtf            out.txt                 unixHelpNotes.rtf
% ^C
[Cameron]@23:19:31 $ exit
exit

Script done on Fri Apr 10 23:19:37 2020
```

```
Script started on 2020-04-13 22:02:56-0700
[22:02:56] wallac21@linux-12:~/Documents/347/csci347_s20$ ./a.out^Hgcc -g -Wall
ush.c
[22:02:59] wallac21@linux-12:~/Documents/347/csci347_s20$ gcc -g -Wall ush.c^H
^H^[[11P./a.out
% echo hello
hello
% ls
a1script  a.out  microshell.c  test.gdb  typescript.pdf  ush.c  ushclone.c
% echo this" "is" "a 123456789xyz
this is a 123456789xyz
% echo this" "is" "a 123456789xyz hello
this is a 123456789xyz hello
% echo this" "is" "a 123456789xyz "hello"
this is a 123456789xyz hello
% echo "hello world"
hello world
% echo "i cant" believe this works
i cant believe this works
% ^C
[22:03:37] wallac21@linux-12:~/Documents/347/csci347_s20$ end

Command 'end' not found, did you mean:

  command 'gnd' from snap gnd (master)
  command 'nd' from deb nd
  command 'ed' from deb ed
  command 'send' from deb mailutils-mh
  command 'send' from deb mmh
  command 'send' from deb nmh
  command 'snd' from deb snd-gtk-jack
  command 'snd' from deb snd-gtk-pulse
  command 'snd' from deb snd-nox
  command 'eid' from deb id-utils
  command 'bnd' from deb bnd
  command 'esd' from deb pulseaudio-esound-compat
  command 'env' from deb coreutils
  command 'and' from deb and
  command 'ent' from deb ent

See 'snap info <snapname>' for additional versions.

[22:03:40] wallac21@linux-12:~/Documents/347/csci347_s20$ quit

Command 'quit' not found, did you mean:

  command 'luit' from deb x11-utils
  command 'quilt' from deb quilt
  command 'qgit' from deb qgit
  command 'quiz' from deb bsdgames
  command 'quot' from deb quota

Try: apt install <deb name>

[22:03:42] wallac21@linux-12:~/Documents/347/csci347_s20$ stop

Command 'stop' not found, but there are 18 similar ones.

[22:03:45] wallac21@linux-12:~/Documents/347/csci347_s20$ ei^H^[[K^H^[[Ksorry
```

```
sorry: command not found
[22:03:57] wallac21@linux-12:~/Documents/347/csci347_s20$ exit
exit

Script done on 2020-04-13 22:03:58-0700
```