

Apr 13, 20 20:20

**title**

Page 1/1

Harry Saliba  
CS 347 Spring 2020

Apr 13, 20 18:10

ush.c

Page 1/4

```

/* CS 352 -- Micro Shell!
 *
 *   Sept 21, 2000,   Phil Nelson
 *   Modified April 8, 2001
 *   Modified January 6, 2003
 *   Modified January 8, 2017
 *
 *   Modified for assignment 1 by Harry Saliba
 *
 */

#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);

/* Shell main */

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

    while (1)
    {
        /* prompt and get line */
        fprintf(stderr, "%s ", "");
        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)

```

Apr 13, 20 18:10

ush.c

Page 2/4

```

{
    pid_t cpid;
    int status;

    /* process the args */
    int argc = 0;
    char **argv = arg_parse(line, &argc);

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

    /* Check for who we are! */
    if (cpid == 0)
    {
        /* We are the child! */
        execvp(argv[0], argv);
        /* execlp returned, 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)
    {
        /* Wait wasn't successful */
        perror("wait");
    }
}

/*Parse Arguments
 * Processes a string containing an executable command into an array of argument
s.
 *
 * @TODO: process command line arguments that have double quotes.
 */
char **arg_parse(char *line, int *argcptr)
{
    /* count argv */
    int count = 0;
    char *pos = line;
    while (*pos != '\0')
    {
        if (*pos == ' ')
        {
            pos++;
        }
        else
        {
            count++;
            while (*pos != '\0' && *pos != ' ')
            {
                if (*pos == '"')

```

```

        {
            pos++;
            while (*pos != '"')
            {
                if (*pos == '\\0')
                {
                    printf("Wrong number of double quotes.\n");
                    return NULL;
                }
                pos++;
            }
            pos++;
        }
    else
    {
        pos++;
    }
}

}

*argcptr = count;

/* malloc size + 1 */
char **argv = (char **)malloc(sizeof(char *) * (count + 1));
argv[count] = NULL;

/* assign pointers and add EOS chars */
count = 0;
pos = line;
while (*pos != '\\0')
{
    if (*pos == ' ')
    {
        pos++;
    }
    else
    {
        argv[count] = pos;
        count++;
        while (*pos != '\\0' && *pos != ' ')
        {
            if (*pos == '"')
            {
                pos++;
                while (*pos != '"')
                {
                    pos++;
                }
                pos++;
            }
            else
            {
                pos++;
            }
        }
        if (*pos != '\\0')
        {
            *pos = '\\0';
            pos++;
        }
    }
}

```

Apr 13, 20 18:10

ush.c

Page 4/4

```
        }
    }
}

/*remove quotes */
count--;
char *nq;
while (count > 0) {
    pos = argv[count];
    nq = argv[count];
    while (*nq != '\0' && *pos != '\0') {
        while (*nq == '"') {
            nq++;
        }
        *pos = *nq;
        pos++;
        nq++;
    }
    *pos = '\0';
    count--;
}

return argv;
}
```

Apr 13, 20 19:15

script-a1

Page 1/1

```
Script started on 2020-04-13 19:12:50-0700
^[]0;harry@harry-pc: ~/cs347/csci347_s20/ush^G^[]01;32mharry@harry-pc^[]00m:^[]0
1;34m~/cs347/csci347_s20/ush^[]00m$ ./ush.o
% echo hello workd
hello workd
% echo " "hello" "world""
hello world
% echo "hello world
Wrong number of double qoutes.
% echo "gottem"he"def cant" figure          this " ' ^H ^H" one out""
gottemhedef cant figure this          one out
% echo thi"s is a s"ingl"e a"rg
this is a single arg
% echo this" "is" "a" "single" "arg
this is a single arg
% echo "this is a single arg"
this is a single arg
% ^C
^[]0;harry@harry-pc: ~/cs347/csci347_s20/ush^G^[]01;32mharry@harry-pc^[]00m:^[]0
1;34m~/cs347/csci347_s20/ush^[]00m$ exit
exit

Script done on 2020-04-13 19:15:01-0700
```