CSCI 352 - UNIX Software Development

Spring 2014 Assignment 1, 100 points

Due: Tuesday, April 8, 2014.

In class we will discuss a mini-shell. (Source code may be found on my web site, http://facultyweb.cs.wwu.edu/~phil/classes/f14/352/minishell.c.) I want you to extend this mini-shell to do simple argument passing to the shell. Specifically, I want you to do the following:

- Read the introduction to UNIX found on the class web page. Even if you already know UNIX. Especially notice the information on the program a2ps(1). I want what you turn in to me to be *at least as nice as what a2ps produces.* Your a2ps output may be two logical pages per physical page.

- Read the man pages on execve(2) and related functions (exec(3)). (Yes, you will need to use a different exec... function than what is currently used in the minishell.)

- Read the man page on malloc and if needed, consult a C programming text on how to use malloc(3) and the C sizeof function.

- Create a working directory for your cs352 work. (Note below the name of the directory for turning in your work. *Do not use the same name as that directory.*) Get a copy of minishell.c and rename it to msh.c. Use msh.c for the additions below.

- Write a function "arg_parse" (Prototype is: char ** arg_parse (char *line)) that returns a pointer to a malloced area that points *into the line parameter* and matches what is needed by execve for the "argv" parameter. The malloced area should be the exact size needed, not too small and not too big. You should call malloc(3) *only once.* Do your processing by breaking up the line on spaces. Do not do more complex processing at this time. You should not copy any characters out of the *line* parameter. You only put zero characters in the *line* parameter and set pointers to characters in the *line* parameter in your malloced area. *Notice* that two or more consecutive spaces is equivalent to a single space. For example, a call to arg_parse with the parameter *line*

```
mycommand     arg1   nextarg arg3 not-arg-5-but-4
```

should return a pointer to an "array" of 6 elements. The first element (index 0) should point to the character 'm' in mycommand, the second element should point to the 'a' of arg1, and so forth. The last element should be a NULL pointer. The characters "mycommand" are not copied to any other storage. The space just after "mycommand" is turned into a zero character so "mycommand" is a zero terminated string. Also, leading and trailing spaces should be ignored! (Before the command name and after the last argument.)

- Modify the mini-shell function processline() to use arg_parse to run commands with arguments. The parent should call arg_parse (before the fork()) and don't forget to free the pointer returned by arg_parse() *in the parent shell process.* (Using execve is not a good idea.)

- Make sure you have no warnings when compiled with "gcc -Wall" on Ubuntu as provided by the department in the labs. (Remember, you have remote access to the linux-01 to linux-12 machines. Domain is cs.wwu.edu. Some kind of ssh client is required to remote access these machines.)

- **Turn in** a printed copy of your code and a "typescript" of a compile and run of your shell. Use script(1) to make the "typescript". (Read the man page.) Remember to use a2ps or something better. I also want a cover page that includes your name, "CS 352 Spring 2014" and the assignment number. You need to do this for ALL your assignments this quarter.

- Make your source code available to me online. This must be done on the CS Lab Ubuntu systems (see above) and your file should be the file "cs352s14/a1/msh.c" relative to your home directory. (UNIX file systems are case sensitive. CS352 is not the same as cs352.) Use chmod(1) to make your home directory readable and searchable by user and other, but **not by group.** The command "chmod 705 /home/username" will do the proper thing for your home directory. Also, do the same thing to your cs352s14 and a1 directories. Make sure your file, "msh.c" is readable by other and it would be best if you made it read only for owner and others. **Do not move or change these files and directories for the entire quarter.** You should not do your development in the cs352s14/a1 directory. The only file in your turn-in directory should be the file "msh.c". The directory is only for turning in your assignment. (You will lose points if there is

more files in that directory than your msh.c source file or if you do not get your permissions set properly.) Use chmod(1) to make your "msh.c" file readable for others.

Note: These are the *only* modifications you are to do to this shell. You must start with my code. Any other kinds of modifications will cause you to lose points. (e.g. Changing the prompt.)

Note 2: I will be sending e-mail from time to time to the class. I will be using your student e-mail address. Make sure that you either read your student e-mail often or have it forwarded to your preferred e-mail address.

Note 3: You should have an account on the CS machines. You need to try to login before class on April 3. If you can't login, send me email and I'll make sure you can login. If your files are not on the server when this assignment is due, it will be considered late. Let me know if you are having problems. Also, e-mails to cs.support@wwu.edu should also get a response.