LINUX SHELL AND PROGRAM PROJECT 1

Project posted at BB:

- Due 01/25 Thu Midnight, currently
- About due date.

- Exam A, 01/31, Wed

UNIX (LINUX) SHELL 1/3

A Unix command line terminal (shell):

```
prompt>
prompt> ls
```

Shell is a user program:

1) output “prompt>”
2) read in from keyboard a string (a command)
3) create a child process
4) The child process will load and exec the string (the command)
5) Parent (the shell) waits for the child to finish, then goes to 1)

UNIX (LINUX) SHELL 2/3

What about

prompt> `ls &` `/* ls will` run in the background */

Shell is a user program:

1) output "prompt>"

2) read in from keyboard a string (a command)

3) create a child process

4) The child process will load and exec the string (a command)

5) Parent and child will run concurrently, i.e., parent (the shell) will not wait for the child to finish, but goes to 1)

UNIX (LINUX) SHELL 3/3

What about

prompt> `ls > list.txt`

/* the output from ls will not show in screen, but save to file list.txt */

Shell is a user program:

1) output "prompt>"
2) read in from keyboard a string (a command)
3) create a child process
4) The child process will
   - the shell closes **standard output**
   - opens the file **`list.txt`**
   - load and exec the string (a command)
5) Parent waits for the child to finish, goes to 1)

SHELL PROJECT (PART I)

```
char *args[MAX LINE/2 + 1]; /* command line arguments */
int should_run = 1; /* flag to determine when to exit program */

while (should_run) {
        printf("osh>"); // Use lastname_pid, instead of "osh"
        fflush(stdout);
        /**
        * After reading user input, the steps are:
        * (1) fork a child process using fork()
        * (2) the child process will invoke execvp()
        * (3) if command included &, parent will NOT invoke wait()
        */
}
```