pzhzqt / **xv6shell-simple**

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also compare across forks.

base: **a2571c0f5dd9b9fb6f362efe6b...** ← compare: **420e7556572bac250de2db81...**

2 commits | **1** file changed | **0** commit comments | **1** contributor

Commits on Jan 26, 2018

pzhzqt 1st version. All required functions worked. d0707d2

Commits on Feb 01, 2018

pzhzqt Added execv to allow executing current folder. Added the part to dete… ⋯ 420e755

Showing **1 changed file** with **41 additions** and **9 deletions**. Unified | Split

50 sh.c

```
15  // All commands have at least a type. Have looked at the type, the code
16  // typically casts the *cmd to some specific cmd type.
17  struct cmd {
18- int type;          //  ' ' (exec), | (pipe), '<' or '>' for redirection
19  };
20
21  struct execcmd {
32  };
33
34  struct pipecmd {
35- int type;          // |
36    struct cmd *left;  // left side of pipe
37    struct cmd *right; // right side of pipe
38  };
61      ecmd = (struct execcmd*)cmd;
62      if(ecmd->argv[0] == 0)
63        _exit(0);
64-     fprintf(stderr, "exec not implemented\n");
65-     // Your code here ...
66      break;
67
68    case '>':
69    case '<':
70      rcmd = (struct redircmd*)cmd;
71-     fprintf(stderr, "redir not implemented\n");
72-     // Your code here ...
73      runcmd(rcmd->cmd);
74      break;
75
78      fprintf(stderr, "pipe not implemented\n");
79      // Your code here ...
80      break;
```

```
15  // All commands have at least a type. Have looked at the type, the code
16  // typically casts the *cmd to some specific cmd type.
17  struct cmd {
18+ int type;          //  ' ' (exec), | or & (pipe), '<' or '>' for redirection
19  };
20
21  struct execcmd {
32  };
33
34  struct pipecmd {
35+ int type;          // | or &
36    struct cmd *left;  // left side of pipe
37    struct cmd *right; // right side of pipe
38  };
61      ecmd = (struct execcmd*)cmd;
62      if(ecmd->argv[0] == 0)
63        _exit(0);
64+       execv(ecmd->argv[0],ecmd->argv);
65+     execvp(ecmd->argv[0],ecmd->argv);
66+       fprintf(stderr, "if this prints out, it means exec failed.\n");
67      break;
68
69    case '>':
70    case '<':
71      rcmd = (struct redircmd*)cmd;
72+     close(rcmd->fd);
73+       open(rcmd->file,rcmd->flags,S_IRUSR|S_IWUSR);
74      runcmd(rcmd->cmd);
75      break;
76
79      fprintf(stderr, "pipe not implemented\n");
80      // Your code here ...
81      break;
```

```
   }
   _exit(0);
 }
 getcmd(char *buf, int nbuf)
 {
   if (isatty(fileno(stdin)))
-    fprintf(stdout, "6.828$ ");
   memset(buf, 0, nbuf);
   if(fgets(buf, nbuf, stdin) == 0)
     return -1; // EOF
   return (struct cmd*)cmd;
 }

 // Parsing

 char whitespace[] = " \t\r\n\v";
-char symbols[] = "<|>";

 int
 gettoken(char **ps, char *es, char **q, char **eq)
   case 0:
     break;
   case '|':

   case '<':
     s++;
     break;
   if(peek(ps, es, "|")){
     gettoken(ps, es, 0, 0);
     cmd = pipecmd(cmd, parsepipe(ps, es));
```

```
+
+  case '&':
+    pcmd = (struct pipecmd*)cmd;
+       if (fork1()==0){
+               runcmd(pcmd->right);
+       }else if (fork1()==0){
+               runcmd(pcmd->left);
+       }else{
+               while(wait(NULL)>0);
+       }
+    break;
   }
   _exit(0);
 }
 getcmd(char *buf, int nbuf)
 {
   if (isatty(fileno(stdin)))
+    fprintf(stdout, "CS450$ ");
   memset(buf, 0, nbuf);
   if(fgets(buf, nbuf, stdin) == 0)
     return -1; // EOF
   return (struct cmd*)cmd;
 }

+struct cmd*
+backcmd(struct cmd *left, struct cmd *right)
+{
+  struct pipecmd* cmd;
+  cmd = malloc(sizeof(*cmd));
+  memset(cmd, 0, sizeof(*cmd));
+  cmd->type = '&';
+  cmd->left = left;
+  cmd->right = right;
+  return (struct cmd*)cmd;
+}
+
 // Parsing

 char whitespace[] = " \t\r\n\v";
+char symbols[] = "<|>&";

 int
 gettoken(char **ps, char *es, char **q, char **eq)
   case 0:
     break;
   case '|':
+  case '&':
   case '<':
     s++;
     break;
   if(peek(ps, es, "|")){
     gettoken(ps, es, 0, 0);
     cmd = pipecmd(cmd, parsepipe(ps, es));
+  }else if(peek(ps, es, "&")){
+       gettoken(ps, es, 0, 0);
+       if(*ps==es){
+               fprintf(stderr, "missing command to parallel\n");
+               exit(-1);
+       }
+       cmd = backcmd(cmd, parsepipe(ps, es));
```

```
273     }
274     return cmd;
275   }
311
312     argc = 0;
313     ret = parseredirs(ret, ps, es);
314   - while(!peek(ps, es, "|")){
315       if((tok=gettoken(ps, es, &q, &eq)) == 0)
316         break;
317       if(tok != 'a') {
```

```
305     }
306     return cmd;
307   }
343
344     argc = 0;
345     ret = parseredirs(ret, ps, es);
346   + while(!peek(ps, es, "|&")){
347       if((tok=gettoken(ps, es, &q, &eq)) == 0)
348         break;
349       if(tok != 'a') {
```

No commit comments for this range