/\*

\* builtin.c : check for shell built-in commands

\* structure of file is

\* 1. definition of builtin functions

\* 2. lookup-table

\* 3. definition of is\_builtin and do\_builtin

\*/

#include <limits.h>

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <pwd.h>

#include <grp.h>

#include <sys/utsname.h>

#include "shell.h"

#include <string.h>

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* builtin function definitions \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

static void bi\_builtin(char \*\* argv); /\* "builtin" command tells whether a command is builtin or not. \*/

static void bi\_cd(char \*\*argv) ; /\* "cd" command. \*/

static void bi\_echo(char \*\*argv); /\* "echo" command. Does not print final <CR> if "-n" encountered. \*/

static void bi\_hostname(char \*\* argv); /\* "hostname" command. \*/

static void bi\_id(char \*\* argv); /\* "id" command shows user and group of this process. \*/

static void bi\_pwd(char \*\* argv); /\* "pwd" command. \*/

static void bi\_quit(char \*\*argv); /\* quit/exit/logout/bye command. \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* lookup table \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

static struct cmd {

char \* keyword; /\* When this field is argv[0] ... \*/

void (\* do\_it)(char \*\*); /\* ... this function is executed. \*/

} inbuilts[] = {

{ "builtin", bi\_builtin }, /\* List of (argv[0], function) pairs. \*/

/\* Fill in code. \*/

{ "echo", bi\_echo },

{ "quit", bi\_quit },

{ "exit", bi\_quit },

{ "bye", bi\_quit },

{ "logout", bi\_quit },

{ "cd", bi\_cd },

{ "pwd", bi\_pwd },

{ "id", bi\_id },

{ "hostname", bi\_hostname },

{ NULL, NULL } /\* NULL terminated. \*/

};

static void bi\_builtin(char \*\* argv) {

/\* Fill in code. \*/

char str[][10]={"echo","quit","exit","bye","logout","cd","pwd","id","hostname"};

int i;

for(i=0;i<9;i++){ //因為使用固定二維陣列，不能用str[i]!＝NULL and if(str[i]==NULL) break;

int number = strcmp(argv[1],str[i]);

if(number == 0){

printf("%s is a builtin feature\n",argv[1]);

break;

}

}

if(i==9){

printf("%s is NOT a builtin feature\n",argv[1]);

}

}

static void bi\_cd(char \*\*argv) {

/\* Fill in code. \*/

if(chdir(argv[1])){

perror("chdir");

exit(1);

}

}

static void bi\_echo(char \*\*argv) {

/\* Fill in code. \*/

int number = strcmp(argv[1],"-n");

if(number == 0){

int a;

a = atoi(argv[2]);

printf("%s\n",argv[a+2]);

}else{

int i;

for(i=1;argv[i]!=NULL;i++)

{

printf("%s ",argv[i]);

}

printf("\n");

}

}

static void bi\_hostname(char \*\* argv) {

/\* Fill in code. \*/

struct utsname uts;

if( uname(&uts) == -1 ) {

perror("myuname.c:main:uname");

exit(1);

}

printf("hostname: %s\n", uts.nodename);

}

static void bi\_id(char \*\* argv) {

/\* Fill in code. \*/

int userid,groupid;

userid = getpid();

groupid = getpgrp();

printf("UserID = %d(chen jia ming),GroupID = %d(chen jia ming)\n",userid,groupid);

}

static void bi\_pwd(char \*\* argv) {

/\* Fill in code. \*/

char \*dir;

long pathmaxlen = pathconf(".", \_PC\_PATH\_MAX);

dir = getcwd((char \*)NULL, pathmaxlen + 1);

if(dir == NULL) {

perror("getcwd");

exit(1);

}

printf("%s\n", dir);

free(dir);

}

static void bi\_quit(char \*\*argv) {

exit(0);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* is\_builtin and do\_builtin \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

static struct cmd \* this; /\* close coupling between is\_builtin & do\_builtin \*/

/\* Check to see if command is in the inbuilts table above.

Hold handle to it if it is. \*/

int is\_builtin(char \*cmd) {

struct cmd \*tableCommand;

for (tableCommand = inbuilts ; tableCommand->keyword != NULL; tableCommand++)

if (strcmp(tableCommand->keyword,cmd) == 0) {

this = tableCommand;

return 1;

}

return 0;

}

/\* Execute the function corresponding to the builtin cmd found by is\_builtin. \*/

int do\_builtin(char \*\*argv) {

this->do\_it(argv);

}