4.BUILT-IN COMMANDS
VE482 LAB 3

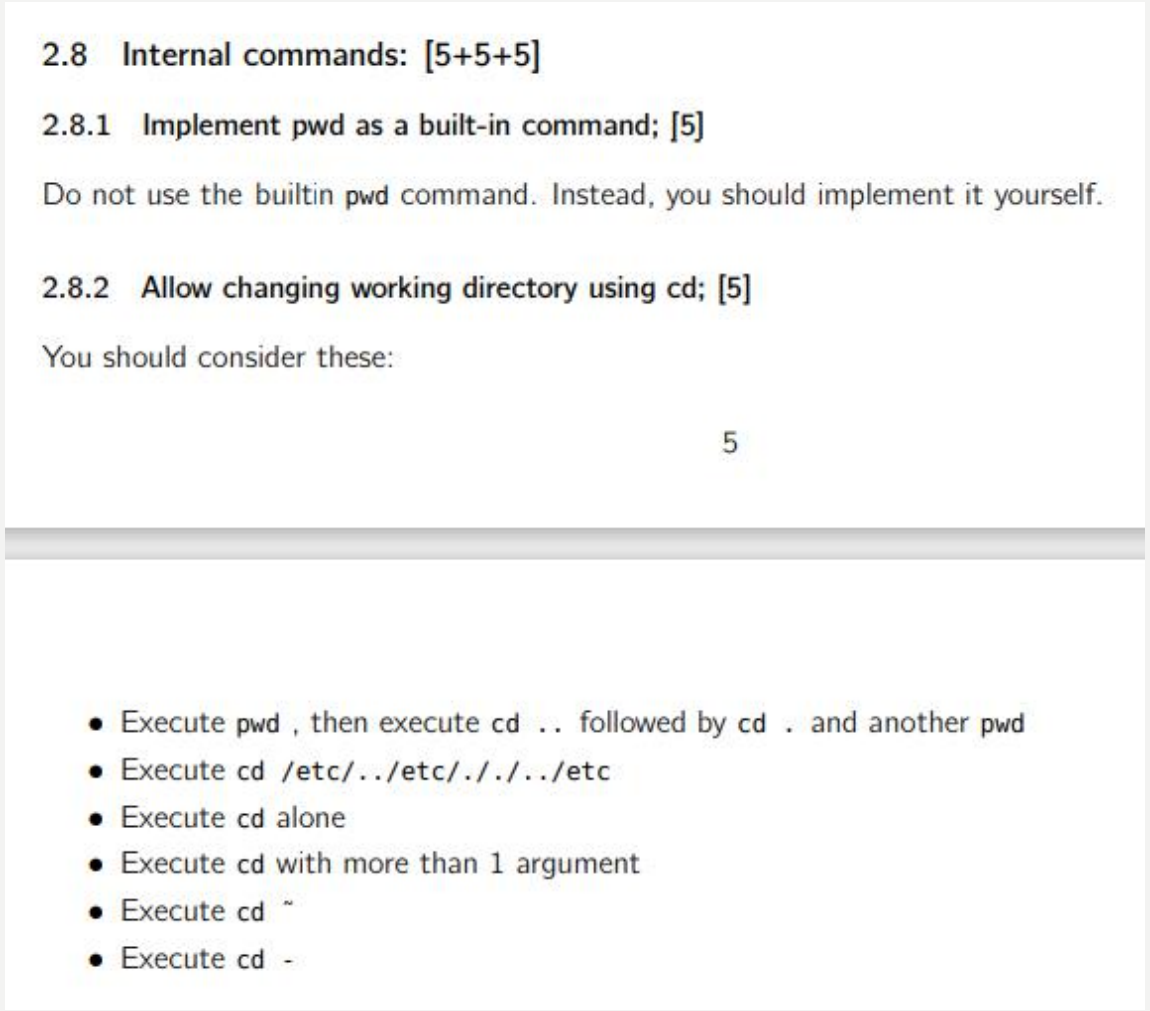
LINYUAN JIANG,
HAOLIN YE,
RUIRONG HUANG,
PENGCHENG XU

SYNOPSIS

bash defines the following built-in commands: :, ., [, alias, bg, bind, break, builtin, case, cd, command, compgen, complete, continue, declare, dirs, disown, echo, enable, eval, exec, exit, export, fc, fg, getopts, hash, help, history, if, jobs, kill, let, local, logout, popd, printf, pushd, pwd, read, readonly, return, set, shift, shopt, source, suspend, test, times, trap, type, typeset, ulimit, umask, unalias, unset, until, wait, while.

Built-in commands in p1

- cd
- pwd



2.8 Internal commands: [5+5+5]

2.8.1 Implement pwd as a built-in command; [5]

Do not use the builtin `pwd` command. Instead, you should implement it yourself.

2.8.2 Allow changing working directory using cd; [5]

You should consider these:

5

- Execute `pwd` , then execute `cd ..` followed by `cd .` and another `pwd`
- Execute `cd /etc/../etc/./././../etc`
- Execute `cd` alone
- Execute `cd` with more than 1 argument
- Execute `cd ~`
- Execute `cd -`

cd [-L|[-P [-e]] [-@]] [dir]

- In Linux '**cd**' (**Change Directory**) command is one of the most important and most widely used command for newbies as well as system administrators. For admins on a headless server, '**cd**' is the only way to navigate to a directory to check log, execute a program/ application/ script and for every other task. For newbie it is among those initial commands they make their hands dirty with.

1. Change from current directory to /usr/local.

```
explcre@LAPTOP-5HQLAHIF: $ cd /usr/local
explcre@LAPTOP-5HQLAHIF:/usr/local$
```

avi@tecmint:/usr/local$

2. Change from current directory to /usr/local/lib using absolute path.

```
explcre@LAPTOP-5HQLAHIF:/usr/local$ cd /usr/local/lib
explcre@LAPTOP-5HQLAHIF:/usr/local/lib$
```

avi@tecmint:/usr/local/lib$

3. Change from current working directory to /usr/local/lib using

```
explcre@LAPTOP-5HQLAHIF:/usr/local$ cd lib
explcre@LAPTOP-5HQLAHIF:/usr/local/lib$
```

avi@tecmint:/usr/local/lib$

4. (a) Move one directory back from where you are now.

```
explcre@LAPTOP-5HQLAHIF:/usr/local/lib$ cd -
/usr/local
explcre@LAPTOP-5HQLAHIF:/usr/local$
```

/usr/local

avi@tecmint:/usr/local$

```
explcre@LAPTOP-5HQLAHIF:/usr/local$ cd ..
explcre@LAPTOP-5HQLAHIF:/usr$
```

avi@tecmint:/usr/local/lib$

- **5. Show last working directory from where we moved (use**

```
explcre@LAPTOP-5HQLAHIF:/usr$ cd --
explcre@LAPTOP-5HQLAHIF:~$ pwd
/home/explcre
```

- **$ cd –**
- **/home/avi**
- **6. Move two directory up from**

```
explcre@LAPTOP-5HQLAHIF:/usr/local$ cd ../../
explcre@LAPTOP-5HQLAHIF:/$ cd ~
```

- **avi@tecmint:/usr/local$ cd ../ ../**

```
explcre@LAPTOP-5HQLAHIF:~$ cd /usr/local
explcre@LAPTOP-5HQLAHIF:/usr/local$ cd ~
explcre@LAPTOP-5HQLAHIF:~$ pwd
/home/explcre
```

- **e directory**
- **from where we**
- **avi@tecmint:/usr/local$ cd ~**

```
explcre@LAPTOP-5HQLAHIF:/usr/local$ cd
explcre@LAPTOP-5HQLAHIF:~$ pwd
/home/explcre
```

- 8. Change working directory to current working directory (seems no

```
explcre@LAPTOP-5HQLAHIF:~/Downloads$ cd .
explcre@LAPTOP-5HQLAHIF:~/Downloads$
```

- avi@tecmint:~/Downloads$

```
explcre@LAPTOP-5HQLAHIF:~/Downloads$ cd ./
explcre@LAPTOP-5HQLAHIF:~/Downloads$
```

- avi@tecmint:~/Downloads$
- 9. Your present working Directory is “/usr/local/lib/python3.4/dist-packages/ ”, change it to “/home/avi/Desktop/ ”, in one line command, by moving up in the directory till ‘/’ then using absolute

How to implement“cd” in c

<unistd.h> header file:

The <unistd.h> header defines miscellaneous symbolic constants and types, and declares miscellaneous functions.It contains the **chdir()** and **fchdir()** functions.

chdir():

The chdir function is used to change the current working directory of the program or process by passing the path to the function as shown in the syntax.

Function declaration: int chdir(const char *pathname);

Return value : The function return a integer value ,it returns 0 if the change of directory was successful otherwise it returns -1 and the current working directory remains unchanged and *errno* is set to to indicate the error type.

Errors:

1.EACCES :Search permission is denied for any component of the pathname.

2.ELOOP :Too many symbolic links were encountered in resolving path.

3.ENAMETOOLONG :The path argument exceeds {PATH_MAX} in length or a pathname component is longer than {NAME_MAX}.

4.ENOENT :A component of path does not name an existing directory or path is an empty string.

5.ENOTDIR :A component of the pathname is not a directory.

Example code:

```
#include<stdio.h>
#include<unistd.h>
int main()
{
    //pass your path in the function
    int ch=chdir("xxx");
    /*if the change of directory was successful it will print successful otherwise
    if(ch<0)
    printf("chdir change of directory not successful\n");
    else
    printf("chdir change of directory successful");
    return 0;
}
```

pwd

BASIC SYNTAX OF PWD:

pwd [OPTION]

OPTIONS USED WITH PWD

Options	Description
-L (logical)	Use PWD from environment, even if it contains symbolic links
-P (physical)	Avoid all symbolic links
–help	Display this help and exit
–version	Output version information and exit

If both '-L' and '-P' options are used, option 'L' is taken into priority. If no option is specified at the prompt, pwd will execute as same as with "-L"

Exit status of command pwd:

0	Success
Non-zero	Failure

1. Different between pwd, pwd –L and pwd –P when symbolic link exist.

```
mike@ubuntu:/var$ ll
总用量 56
drwxr-xr-x 14 root root     4096 8月   5  2019 ./
drwxr-xr-x 24 root root     4096 10月 23  2019 ../
drwxr-xr-x  2 root root     4096 9月  19 06:38 backups/
drwxr-xr-x 16 root root     4096 9月   6  2019 cache/
drwxrwsrwt  2 root whoopsie 4096 9月  28 00:24 crash/
drwxr-xr-x 65 root root     4096 9月  18 18:49 lib/
drwxrwsr-x  2 root staff    4096 4月  24  2018 local/
lrwxrwxrwx  1 root root        9 9月   6  2019 lock -> /run/lock/
drwxrwxr-x 12 root syslog   4096 9月  30 01:21 log/
drwxrwsr-x  2 root mail     4096 8月   5  2019 mail/
drwxrwsrwt  2 root whoopsie 4096 8月   5  2019 metrics/
drwxr-xr-x  2 root root     4096 8月   5  2019 opt/
lrwxrwxrwx  1 root root        4 9月   6  2019 run -> /run/
drwxr-xr-x 14 root root     4096 9月  27 06:55 snap/
drwxr-xr-x  7 root root     4096 8月   5  2019 spool/
drwxrwxrwt  9 root root     4096 9月  28 17:05 tmp/
mike@ubuntu:/var$ cd lock/
mike@ubuntu:/var/lock$ pwd
/var/lock
mike@ubuntu:/var/lock$ pwd -P
/run/lock
mike@ubuntu:/var/lock$ pwd -L
/var/lock
```

How to implement “pwd” in c

getcwd():

The getcwd() function places an absolute pathname of the current working directory in the array pointed to by buf, and returns buf. The size argument is the size in bytes of the array pointed to by the buf argument. If buf is a null pointer, the behaviour of getcwd() is undefined. The return value represent our current working directory.

```
char *getcwd(char *buf, size_t size);
```

Function return: The getcwd() function returns a pointer which points to a character array where the path of current working directory is stored. In case the path is not found then it returns a null pointer and the contents of the array are undefined and the errno is set to indicate the type of error.

Type of errors in getcwd():

1.**EINVAL**:The size argument is 0.

2.**ERANGE**:The size argument is greater than 0, but is smaller than the length of the pathname +1.

3.**EACCES**:Read or search permission was denied for a component of the pathname.

4.**ENOMEM**:Insufficient storage space is available.

Example code:

```c
#include <unistd.h>
#include <stdio.h>
int main() {
  char cwd[256];


    if (getcwd(cwd, sizeof(cwd)) == NULL)
      perror("getcwd() error");
    else
      printf("current working directory is: %s\n", cwd);

   return 0;

}
```