【Linux Programming】Day18

Tags	
Date	@June 10, 2022
Summary	Temporary Files and User Information

【Ch4】Work with Files

4.4 Temporary Files

We need to ensure that the applications choose a unique filename to use for the temporary file.

```
#include <stdio.h>

char *tmpnam(char *s);
```

The `tmpnam` function returns a valid filename that isn't the same as any existing file.

If the string `s` isn't null, the filename will also be written to it. The string is assumed to be at least `L_tmpnam`(usually 20) characters long.

If the temporary file is to be used immediately, we can name it and open it at the same time using the `tmpfile` function.

Note: This is important because another program could create a file with the same name as that returned by `tmpnam`.

```
#include <stdio.h>

FILE *tmpfile(void);
```

The `tmpfile` returns a stream pointer that refers to a unique temporary file. The file is opened for reading and writing, and it will be automatically deleted when all references to the file are closed.

```
#include <stdio.h>
#include <stdlib.h>

int main(int agc, char** argv) {
  char *tmp_file_name = tmpnam(NULL);
  printf("Tmp file created: %s\n", tmp_file_name);

  FILE *tmp_fp = tmpfile();
  if (tmp_fp)
    printf("Tmp file created successfully.\n");
  return 0;
}
```

4.5 User Information

When a user logs in to a Linux system, he has a username and password. Once these have been validated, the user is presented with a shell. The user also has a unique user identifier known as a UID.

The UID has its own type-`uid_t`- defined in `sys/types.h`.

```
#include <sys/types.h>
#include <unistd.h>

uid_t getuid(void);
char *getlogin(void);
```

The `getuid` function returns the UID with which the program is associated. This is usually the UID of the user who started the program.

The `getlogin` function returns the login name associated with the current user.

The system file `/etc/passwd` contains a database dealing with user accounts. It consists of lines, one per user, that contain the username encrypted password, user identifier(UID), group identifier(GID), full name, home directory, and default shell.

`neil:zBqxfqedfpk:500:100:Neil Matthew:/home/neil:/bin/bash`

```
#include <sys/types.h>
#include <pwd.h>

struct passwd *getpwuid(uid_t uid);
struct passwd *getpwnam(const char *name);
```

passwd Member	Description
`char *pw_name`	The user's login name
`uid_t pw_uid`	The UID number
`gid_t pw_gid`	The GID number
`char *pw_dir`	The user's home directory
`char *pw_gecos`	The user's full name
`char *pw_shell`	The user's default shell