

Review

- `link()`, `unlink()` System Call
- `remove()`, `rename()` System Call
- Symbolic Links
 - Symbolic link to directory
 - Symbolic link to a executable file
- `symlink()` System Call
- File Times
- `utime()` System Call
- `mkdir()` and `rmdir()` System Call
- `chdir()` Systemcall
- Standard I/O Libraries

Preview

- System Data Files
 - Password File
 - Shadow Password File
 - Group Password File
 - Other System Data Files
 - Hosts
 - Networks
 - Protocols
 - Services
 - Login Account
 - Time and Date

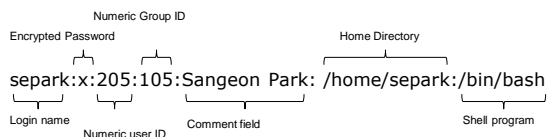
System Data Files (Password File)

- There are several data files required for normal operation:
 - The password file `/etc/passwd`
 - The group file `/etc/group`
- Two files are used as a system data.
Ex) `/etc/passwd` is used every time a user log in to the Linux system.

System Data Files (Password File)

- The Linux password file contains the fields. It is defined in `<pwd.h>`
- Historically the password file has been stored in `/etc/passwd` and has been an ASCII file.
- When we log in to a Linux system, we enter login name, followed by a password.
- The system look up a login name in its password file located in `/etc/passwd`
- Each line usually contains the seven fields separated by colons.
- But number of fields depends on system

System Data Files (Password File)



System Data Files (Password File)

```
#include <pwd.h>

struct passwd {
    char *pw_name; /*User's login name*/
    char *pw_passwd; /* Encrypted password, not available*/
    uid_t pw_uid; /*Numerical user ID*/
    gid_t pw_gid; /* Numerical Group ID */
    time_t pw_change; /* Passwd changed time, not available*/
    char *pw_class; /* user access class, not available*/
    char *pw_gecos; /* user's full name , and other info.*/
    char *pw_dir; /* user's login directory */
    char *pw_shell; /* user's login shell */
    time_t pw_expire; /* passwd expiration time */
};
```

System Data Files (Password File)

- Linux define two functions to get entries from the password file.

```
#include <sys/types.h>
#include <pwd.h>
struct passwd *getpwuid(uid_t uid);
struct passwd *getpwnam(const char *name);
```

- These functions are used to loop up an entry given a user's login name or numerical user ID

System Data Files (Password File)

- The function `getpwuid()` is used by the `ls` program.
- The function `getpwnam()` is used by the login program, when a user enter his/her login name.

```
/*printpasswd.c print out file's infomations*/
#include <sys/types.h>
#include <sys/stat.h>
#include <pwd.h>
#include <stdio.h>
void err_sys(char *str)
{
    printf("%s\n",str);
    exit (1);
}
int main(int argc, char *argv[])
{
    struct stat statbuf;
    struct passwd *pwd;
    if (argc != 2)
        err_sys (" argument number error ");
    if (stat (argv[1], &statbuf) < 0) /*get a usefule data from l-node */
        err_sys("stat error for foo");
    /*need pass user id for getting information from passwd*/
    if ((pwd = getpwuid(statbuf.st_uid)) != NULL)
    {
        printf("Owner's login name: %s", pwd->pw_name);
        printf("Owner's passwd: %s \n", pwd->pw_passwd);
        printf("Owner's Numerical user ID: %d \n", pwd->pw_uid);
        printf("Owner's Numerical group ID: %d \n", pwd->pw_gid);
        printf("Owner's login directory: %s", pwd->pw_dir);
        printf("Owner's login shell: %s \n", pwd->pw_shell);
        printf("Owner's full Name: %s", pwd->pw_gecos);
    }
    else
        printf("fd", statbuf.st_uid);
    exit (0);
}
```

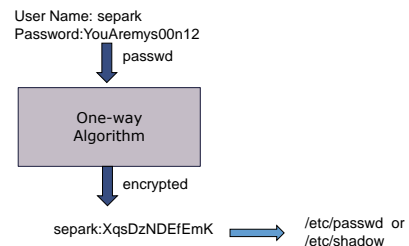
Password Security and Encryption

- Most of the more recent Linux distributions include `passwd` programs that do not allow you to set a easily guessable password.
- Most Unix like system, use a **one-way encryption algorithm**, called DES (Data Encryption Standard) to encrypt your passwords.

Password Security and Encryption

- Once a password is encrypted with **one-way encryption algorithm**, there is no way convert to plain text password.
- This encrypted password is then stored in `/etc/passwd` or in `/etc/shadow`.
- When you attempt to login, the password you type in is encrypted again and compared with the entry in the file that stores your passwords.
- If they match, it must be the same password, and you are allowed access.

Password Security and Encryption



System Data Files (Shadow Password)

- To make it harder to obtain the encrypted password, Linux store the encrypted password in /etc/shadow.
- The shadow file contains **user name encrypted password** and other information related to password.
- It is readable only by super user.
- This shadow file can be accessed by few system calls such as **login()** for login and **passwd()** for change password.
-

COSC350 System Software, Fall 2020
Dr.Sang-Eon Park

13

System Data Files (Group file)

- The group file is located in /etc/group.
- It contains the information for each group.

```
#include <grp.h>

struct group {
    char *gr_name; /* Group name */
    char *gr_passwd; /* Encrypted password. */
    gid_t gr_gid; /* Group ID. */
    char **gr_mem; /* List of group members */
};
```

COSC350 System Software, Fall 2020
Dr.Sang-Eon Park

14

System Data Files (Group file)

- Linux define two functions to get entries from the group file.

```
#include <sys/types.h>
#include <grp.h>
struct group *getgrgid(uid_t uid);
struct group *getgrnam(const char *name);
```

COSC350 System Software, Fall 2020
Dr.Sang-Eon Park

15

```
/*printgroup.c print out file's group information*/
#include <sys/types.h>
#include <sys/stat.h>
#include <grp.h>
#include <stdio.h>

void err_sys(char *str)
{
    printf("%s\n",str);
    exit (1);
}

int main(int argc, char *argv[])
{
    struct stat statbuf;
    struct group *gr;
    if (argc != 2)
        err_sys (" argument number error ");
    if (stat (argv[1], &statbuf) < 0) /*get stat data from l-node */
        err_sys("stat error for lnode");

    if ((gr = getgrgid(statbuf.st_gid)) != NULL)
    {
        printf("Group name: %s\n", gr->gr_name);
        printf("Group number: %d\n", gr->gr_gid);
        while (gr->gr_mem[i] != NULL)
        {
            printf("%s \n",gr->gr_mem[i] );
            i++;
        }
    }
    else
        printf("%d", statbuf.st_gid);

    exit (0);
}
```

Dr.Sang-Eon Park

Other System Data Files

- Hosts
 - Data file: /etc/hosts
 - Header: <netdb.h>
 - Structure: hostent
 - Functions: gethostbyname, gethostbyaddr
- Networks
 - Data file: /etc/networks
 - Header: <netdb.h>
 - Structure: netent
 - Functions: getnetbyname, getnetbyaddr

COSC350 System Software, Fall 2020
Dr.Sang-Eon Park

17

Other System Data Files

- Protocols
 - Data File: /etc/protocols
 - Header: <netdb.h>
 - Structure: protoent
 - Functions: getprotobyname, getprotobynumber
- Services:
 - Data File: /etc/services
 - Header: <netdb.h>
 - Structure: servent
 - Functions: getservbyname, getservbyport

COSC350 System Software, Fall 2020
Dr.Sang-Eon Park

18

Login Account

- Unix like system, including Linux, provide two system data files regarding login account.
 - /etc/utmp file – keeps track of all the users currently logged in (in Linux: **/var/run/utmp**)
 - /usr/adm/wtmp file – keeps track of all logins and logouts. (in Linux: **/var/log/wtmp**)
- One type of record is written to both files, a binary record consist of following structure.

Login Account

```
struct utmp {
    char ut_line[8]; /* tty name (name of terminal on standard input) */
    char ut_name[8]; /* user name */
    long ut_time; /* time on */
    ...
};
```

- On login, one of this structures was filled in and written to the **utmp** file by **login** program.
- On logout, the entry in the **utmp** file was erased by **init** program and a new entry is appended to the **wtmp** file.
- The **who** program read the utmp file and printed its contents in a readable form.

Time and Date Routine

- The basic time service provided by the Linux kernel is to count the number of seconds that has passed since the Epoch:00:00:00 Jan. 1 1970 (time t).
- The time function returns the current time and date. time_t is a integer type

```
#include <time.h>
time_t time (time_t *calpnr);
```

Time and Date Routine

- The two functions localtime and gmtime convert a calendar time into a broken-down time structure.

```
#include <time.h>

struct tm *gmtime(const time_t *timer); /* convert to UTC */

struct tm *localtime(const time_t *timer); /* convert to local time */
```

Coordinated Universal Time (abbreviated **UTC**) the primary time standard by which the world regulates clocks and time. It is used for servers for online service.

Time and Date Routine

```
struct tm {
    int tm_sec; /* seconds [0,61] */
    int tm_min; /* minutes [0,59] */
    int tm_hour; /* hour [0,23] */
    int tm_mday; /* day of month [1,31] */
    int tm_mon; /* month of year [0,11] */
    int tm_year; /* years since 1900 */
    int tm_wday; /* day of week [0,6] (Sunday = 0) */
    int tm_yday; /* day of year [0,365] */
    int tm_isdst; /* daylight savings flag */
};
```

Time and Date Routine

- The asctime and ctime function produce the 26 byte string.
Tue Sep 23 07:07:21 2014

```
#include <time.h>
char *asctime(const struct tm *timeptr);
char *ctime(const time_t *clock);
```

Time and Date Routine

```

/*displaytime.c display current date and time */
#include <stdio.h>
#include <time.h>

int main()
{
    time_t t1;

    t1 = time((time_t *)0);
    printf("current time is: %s\n",asctime(localtime(&t1)));
    exit (0);
}

```

Time and Date Routine

```

/* timer.c */
#include <stdio.h>
#include <sys/types.h>
#include <time.h>

int main()
{
    int i;
    time_t t1;

    for (i=1;i<=10;i++)
    {
        t1 = time((time_t *)0);
        printf("The time passed %d seconds since Jan.1 1971\n", t1);
        // asctime output is formatted time 26 byte string Tue Sep 23 07:07:21 2014
        printf("current time is: %s\n",asctime(localtime(&t1)));
        sleep(1);
    }

    exit (0);
}

```