

System Information: Chapter 6

☐ Users -- /etc/passwd

- ☐ Root -- uid 0

- ☐ encrypted passwords, shadow passwords

- ☐ finger(1) (Print out password information)

- ☐ chfn(1) (Change information in password file)

- ☐ chsh(1) (Change your login shell)

☐ Groups -- /etc/group

- ☐ wheel (BSD) - gid 0

☐ Program access?

- ☐ struct passwd *getpwent(void); (open, get next)

- ☐ void setpwent(void); (rewind)

- ☐ void endpwent(void); (close)

- ☐ struct passwd *getpwnam(const char *login);

- ☐ struct passwd *getpwuid(uid_t uid);

struct passwd

```
struct passwd { /* Linux version */
    char  *pw_name;    /* user name */
    char  *pw_passwd;  /* encrypted password */
    uid_t pw_uid;      /* user uid */
    gid_t pw_gid;      /* user gid */
    char  *pw_gecos;   /* general information */
    char  *pw_dir;     /* home directory */
    char  *pw_shell;   /* default shell */
};
```

Returned pointers are not dynamically allocated
Don't free them!

Group information

- ❑ struct group *getgrent(void); (open, get next)
- ❑ void setgrent(void); (rewind)
- ❑ void endgrent(void); (close)
- ❑ struct group *getgrnam(const char *name);
- ❑ struct group *getgrgid(gid_t gid);

```
struct group {  
    char *gr_name;    /* group name */  
    char *gr_passwd;  /* group password */  
    gid_t gr_gid;     /* group id */  
    char **gr_mem;    /* group members */  
};
```

Supplementary Group Ids

- ❑ Belong to more than just the primary group
- ❑ Group checks -- if you are a member of that group
- ❑ User can change group of any file/directory to any group of which they are a member.

Other data & configuration files

Very UNIX version specific

Network related

- ☐ /etc/hosts
- ☐ /etc/networks
- ☐ /etc/protocols
- ☐ /etc/services
- ☐ /etc/resolv.conf
- ☐ /etc/ssh

Shell related

- ☐ /etc/csh.cshrc
- ☐ /etc/profile

Other data & configuration files (page 2)

System startup

□ /etc/rc* (BSD)

□ /etc/init.d (SYSV)

Generic information

□ /etc/termcap (/usr/share/misc/termcap)

□ /etc/printcap

□ /etc/motd

Many more

Login accounting & such

finger(1)

- ☐ List information about users

who(1)

- ☐ /var/log/wtmp

- ☐ /var/run/utmp

last(1)

uptime(1)

rwwho(1), ruptime(1)

lastcomm(1) [accton(8)]

System identification

uname(1)

uname(3)

```
#define _SYS_NMLN    256
```

```
#if !defined(_POSIX_C_SOURCE) && !defined(_XOPEN_SOURCE)
```

```
#define SYS_NMLN      _SYS_NMLN
```

```
#endif
```

```
struct utsname {
```

```
    char  sysname[_SYS_NMLN];    /* Name of this OS. */
```

```
    char  nodename[_SYS_NMLN];    /* Name of this network node. */
```

```
    char  release[_SYS_NMLN];    /* Release level. */
```

```
    char  version[_SYS_NMLN];    /* Version level. */
```

```
    char  machine[_SYS_NMLN];    /* Hardware type. */
```

```
};
```

BSD -- gethostname(3)

Date & Time routines

`time_t time(time_t *calptr);`

□ seconds since epoch

□ midnight Jan 1, 1970

BSD systems / Linux systems

□ `int gettimeofday(struct timeval *tp, struct timezone *tzp);`

□ `int settimeofday(struct timeval *tp, struct timezone *tzp);`

`struct timeval {`

`long tv_sec; /* seconds since Jan. 1, 1970 */`

`long tv_usec; /* and microseconds */`

`};`

`struct timezone {`

`int tz_minuteswest; /* of Greenwich */`

`int tz_dsttime; /* type of dst correction to apply */`

`};`

Other time related stuff

Most UNIX systems have hardware clock set to Greenwich.

Time zone information (Actual details vary, NetBSD in example)

□ /etc/localtime

□ TZ=/usr/share/zoneinfo/US/Alaska date

time related functions

□ strftime(3)

□ ctime(3) and related functions

□ struct tm *localtime(const time_t *timep)

□ char *asctime(const struct tm *tm)

struct tm {

int tm_sec; /* Seconds (0-60) */

int tm_min; /* Minutes (0-59) */

int tm_hour; /* Hours (0-23) */

int tm_mday; /* Day of the month (1-31) */

int tm_mon; /* Month (0-11) */

int tm_year; /* Year - 1900 */

int tm_wday; /* Day of the week (0-6, Sunday = 0) */

int tm_yday; /* Day in the year (0-365, 1 Jan = 0) */

int tm_isdst; /* Daylight saving time */

};

