

Advanced Programming in the UNIX Environment

**Week 04, Segment 6:
`getpwuid(2)`, `/etc/groups`**

**Department of Computer Science
Stevens Institute of Technology**

Jan Schaumann

`jschauma@stevens.edu`

`https://stevens.netmeister.org/631/`

/etc/passwd

```
root:*:0:0:Charlie &:/root:/bin/csh
toor:*:0:0:Bourne-again Superuser:/root:/rescue/sh
daemon:*:1:1:The devil himself:/sbin/nologin
operator:*:2:5:System &/usr/guest/operator:/sbin/nologin
bin:*:3:7:Binaries Commands and Source:/sbin/nologin
games:*:7:13:& pseudo-user:/usr/games:/sbin/nologin
postfix:*:12:12:& pseudo-user:/var/spool/postfix:/sbin/nologin
named:*:14:14:& pseudo-user:/var/chroot/named:/sbin/nologin
ntpd:*:15:15:& pseudo-user:/var/chroot/ntpd:/sbin/nologin
sshd:*:16:16:& pseudo-user:/var/chroot/sshd:/sbin/nologin
uucp:*:66:1:UNIX-to-UNIX Copy:/nonexistent:/sbin/nologin
nobody:*:32767:39:Unprivileged user:/nonexistent:/sbin/nologin
jschauma:*:1000:100:Jan Schaumann,Lieb Building,555-1234,555-2233:/home/jschauma:/bin/sh
fred::1001:100:::/bin/sh
drwho:*:1003:100:The Doctor:/home/drwho:/bin/date
alice:*:1004:100::/home/alice:
alice:*:1002:100::/home/alice:/bin/ksh
```

getpwuid(2) and friends

```
#include <pwd.h>

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

Returns: pointer on success, NULL on error

```
struct passwd {
    char    *pw_name;          /* user name */
    char    *pw_passwd;        /* hashed 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 */
};
```

getpwuid(3)

The `master.passwd` file is readable only by root, and consists of newline separated ASCII records, one per user, containing ten colon ":" separated fields.

Each line has the form:

`name:password:uid:gid:class:change:expire:gecos:home_dir:shell`

These fields are as follows:

hashed, actually

<u>name</u>	User's login name.
<u>password</u>	User's <u>encrypted</u> password.
<u>uid</u>	User's id.
<u>gid</u>	User's login group id.
<u>class</u>	User's login class.
<u>change</u>	Password change time.
<u>expire</u>	Account expiration time.
<u>gecos</u>	General information about the user.
<u>home_dir</u>	User's home directory.
<u>shell</u>	User's login shell.

Be aware that each line is limited to 1024 characters; longer ones will be ignored. This limit can be queried through `sysconf(3)` by using the `_SC_GETPW_R_SIZE_MAX` parameter.

The `passwd` file is generated from the `master.passwd` file by `pwd_mkdb(8)`, has the class, change, and expire fields removed, and the password field replaced by a "*".



```
[jschauma@apue$ ls -l
total 40
-rwx----- 1 jschauma    users   931 Sep 21 00:25 cd.c
-rw-r--r--  1 jschauma    users  3673 Sep 25 01:50 getpw.c
-rw-r--r--  1 jschauma    users 3881 Sep 25 02:59 groups.c
-rw-r--r--  1 jschauma    users  471 Sep 20 02:56 lns.c
-rw-r--r--  1 jschauma    users  460 Sep 20 02:42 rename.c
-rw-----  1 jschauma    users 1337 Sep  3 2019 sizeof.c
-rwx----- 1 jschauma    users 2430 Sep 20 14:26 wait-unlink.c
jschauma@apue$ ]
```



```
[jschauma@apue$ ls -l
```

```
total 40
```

```
-rwx----- 1 jschauma users 931 Sep 21 00:25 cd.c
-rw-r--r-- 1 jschauma users 3673 Sep 25 01:50 getpw.c
-rw-r--r-- 1 jschauma users 3881 Sep 25 02:59 groups.c
-rw-r--r-- 1 jschauma users 471 Sep 20 02:56 lns.c
-rw-r--r-- 1 jschauma users 460 Sep 20 02:42 rename.c
-rw----- 1 jschauma users 1337 Sep 3 2019 sizeof.c
-rwx----- 1 jschauma users 2430 Sep 20 14:26 wait-unlink.c
```

```
[jschauma@apue$ ls -ln
```

```
total 40
```

```
-rwx----- 1 1000 100 931 Sep 21 00:25 cd.c
-rw-r--r-- 1 1000 100 3673 Sep 25 01:50 getpw.c
-rw-r--r-- 1 1000 100 3881 Sep 25 02:59 groups.c
-rw-r--r-- 1 1000 100 471 Sep 20 02:56 lns.c
-rw-r--r-- 1 1000 100 460 Sep 20 02:42 rename.c
-rw----- 1 1000 100 1337 Sep 3 2019 sizeof.c
-rwx----- 1 1000 100 2430 Sep 20 14:26 wait-unlink.c
```

```
jschauma@apue$
```

getpwuid(2) and friends

```
jschauma@apue$ head -2 /etc/passwd
root:*:0:0:Charlie &:/root:/bin/csh
toor:*:0:0:Bourne-again Superuser:/root:/rescue/sh
jschauma@apue$ head -2 /etc/master.passwd
head: /etc/master.passwd: Permission denied
jschauma@apue$ sudo head -2 /etc/master.passwd
root:$sha1$19739$syqYhzyU$SbJUFaglcumeQBg6hk5uthIR/83H:0:0::0:0:Charlie
&:/root:/bin/csh
toor:$sha1$20339$p1vGuM0s$RBLfoz9mFTr4gJDMywdXwVZKhadH:0:0::0:0:Bourne-
again Superuser:/root:/rescue/sh
jschauma@apue$
```

getpwuid(2) and friends

```
#include <shadow.h>
```

```
struct spwd *getspnam(char *name);
```

Returns: pointer on success, NULL on error

```
struct spwd {  
    char      *sp_namp;          /* user login name */  
    char      *sp_pwdp;          /* encrypted password */  
    long int   sp_lstchg;        /* last password change */  
    long int   sp_min;           /* days until change allowed. */  
    long int   sp_max;           /* days before change required */  
    long int   sp_warn;          /* days warning for expiration */  
    long int   sp_inact;          /* days before account inactive */  
    long int   sp_expire;         /* date when account expires */  
    unsigned long int  sp_flag;  /* reserved for future use */  
}
```

getpwuid(2) and friends

```
#include <pwd.h>
```

```
struct passwd *getpwuid(uid_t uid);
```

```
struct passwd *getpwnam(const char *name);
```

Returns: pointer on success, NULL on error

```
#include <pwd.h>
```

```
struct passwd *getpwent(void);
```

Returns: pointer on success, NULL on error

```
void setpwent(void);
```

```
void endpwent(void);
```

```
[jschauma@apue$ ./a.out toor
toor:*:0:0:Bourne-again Superuser:/root:/rescue/sh
[jschauma@apue$ ./a.out alice
alice:*:1004:100::/home/alice:
[jschauma@apue$ grep alice /etc/passwd
alice:*:1004:100::/home/alice:
alice:*:1002:100::/home/alice:/bin/ksh
[jschauma@apue$ ./a.out avocado
a.out: avocado: No such user
[jschauma@apue$ ./a.out 555
a.out: 555: No such user
[jschauma@apue$ ./a.out jschauma
jschauma:*:1000:100:Jan Schaumann,Lieb Building,555-1234,555-2222:/home/jschauma
:/bin/sh
[jschauma@apue$ sudo ./a.out jschauma
jschauma:$sha1$22781$uwM2CnsW$va3dRmWmwAH1x1TYRUWDjjjsVY0Tc:1000:100:Jan Schauman
n,Lieb Building,555-1234,555-2222:/home/jschauma:/bin/sh
[jschauma@apue$ ./a.out fred
fred:*:1001:100:::/bin/sh
[jschauma@apue$ grep fred /etc/passwd
fred::1001:100:::/bin/sh
[jschauma@apue$ sudo ./a.out fred
fred::1001:100:::/bin/sh
jschauma@apue$
```

/etc/group

Called a *group database* by POSIX and usually found in /etc/group, the group file contains the following fields:

Description	struct passwd member	POSIX.1
group name	char *gr_name	✓
hashed password	char *gr_passwd	
numerical GID	gid_t gr_gid	✓
array of pointers to usernames	char **gr_mem	✓

the user in the group database.

If the user is not a member of the specified group, and the group requires a password, the user will be prompted for the group password.

```
[jschauma@apue$ newgrp guest
[Password:
[jschauma@apue$ groups
guest users wheel
[jschauma@apue$ cat file
Only members of the group 'guest' can read this.
Well, and root, of course.
[jschauma@apue$ grep guest /etc/group
guest:$sha1$22781$uwM2CnsW$va3dRmWmwAH1x1TYRUWDjjsVY0Tc:31:root
[jschauma@apue$ grep games /etc/group
games::13:
[jschauma@apue$ newgrp games
newgrp: Sorry
[jschauma@apue$ groups
guest users wheel
[jschauma@apue$ exit
[jschauma@apue$ groups
guest users wheel
jschauma@apue$ ]
```

getpwuid(2) and friends

```
#include <grp.h>
```

```
struct group *getgrgid(gid_t gid);
```

```
struct group *getpgrnam(const char *name);
```

Returns: pointer on success, NULL on error

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

getpwuid(2) and friends

```
#include <grp.h>
```

```
struct group *getgrgid(gid_t gid);
```

```
struct group *getpgrnam(const char *name);
```

Returns: pointer on success, NULL on error

```
#include <grp.h>
```

```
struct group *getgrent(void);
```

Returns: pointer on success, NULL on error

```
void setgrent(void);
```

```
void endgrent(void);
```

getpwuid(2) and friends

```
#include <unistd.h>
```

```
int getgroups(int gidsetlen, gid_t *gidset);
```

Returns: number of groups on success, -1 on error

```
int getgrouplist(const char *name, gid_t basegid, gid_t *groups, int *ngroups);
```

Returns: 0 on success, -1 if groups is too small

- POSIX does not define whether `getgroups(2)` includes the egid
- calling `getgroups(2)` with `gidsetlen == 0` leaves `gidset` unmodified
- `getgrouplist(3)` is non-standard

```
[jschauma@apue$ vim groups.c
[jschauma@apue$ cc groups.c
[jschauma@apue$ ./a.out
users wheel
[jschauma@apue$ ./a.out root
wheel kmem sys tty operator staff guest nvmm
[jschauma@apue$ grep jschauma /etc/group
wheel:*:0:root,jschauma
[jschauma@apue$ newgrp guest
>Password:
[jschauma@apue$ ./a.out
guest users wheel
[jschauma@apue$ ^D
[jschauma@apue$ ./a.out
users wheel
[jschauma@apue$ ls -li `which id` `which groups`
1885850 -r-xr-xr-x 3 root wheel 19304 Feb 14 2020 /usr/bin/groups
1885850 -r-xr-xr-x 3 root wheel 19304 Feb 14 2020 /usr/bin/id
jschauma@apue$ ]
```

Other system databases

Similar routines as for password/group for accessing system data files:

Description	Data File	Structure	Lookup functions
hosts	/etc/hosts	hostent	gethostname(3) gethostbyaddr(3)
networks	/etc/networks	netent	getnetbyname(3) getnetbyaddr(3)
protocols	/etc/protocols	portent	getprotobynumber(3) getprotobynumber(3)
services	/etc/services	servent	getservbyname(3) getservbyport(3)

getpw* and /etc/group

- `getpw*` is pretty straight forward
- only euid 0 gets the password hash from `/etc/master.passwd`
- see `getspnam(3)` / `getspent(3)` on systems using `/etc/shadow`
- although rarely seen, groups can have a password; use `newgrp(1)` to gain access
- supplementary and primary groups make iterating over group membership annoyingly complex
- We'll revisit other `getFOObyBAR` functions in future lectures.
- Coming up next: Fun Times with File Times!