NAME

getpwent, getpwnam, getpwuid, setpwent, endpwent, setpwfile − password file routines

SYNOPSIS

#include <pwd.h>

struct passwd *getpwent(void)
struct passwd *getpwnam(const char ***_name_)**
struct passwd *getpwuid(uid_t _uid_**)**
int setpwent(void)
void endpwent(void)
void setpwfile(const char ***_file_)**

DESCRIPTION

These functions are used to obtain information from the password file. They return this information in a **struct passwd** as defined by <pwd.h>:

```
struct passwd {
    char    *pw_name;       /* login name */
    char    *pw_passwd;     /* encrypted password */
    uid_t   pw_uid;         /* numeric user id */
    gid_t   pw_gid;         /* numeric group id */
    char    *pw_gecos;      /* user full name and other info */
    char    *pw_dir;        /* user's home directory */
    char    *pw_shell;      /* name of the user's shell */
};
```

Getpwent() reads the password file entry by entry. **Getpwnam()** scans the entire password file for the user with the given _name_. **Getpwuid()** looks for the first user with the given _uid_. The **setpwent()** and **endpwent()** functions are used to open and later close the password file. With **setpwfile()** one can specify the file to read other than the normal password file. This only sets the name, the next **setpwent()** call will open the file. Do not touch the file name while it is active. Use **setpwfile(NULL)** to revert back to the normal password file.

The usual way to scan the password file is (error checking omitted):

```
setpwent();
while ((pw = getpwent()) != NULL)
        if (appropriate_test(pw)) break;
endpwent();
```

The **pw** variable contains the entry that is wanted if non-NULL. The **getpwnam()** and **getpwuid()** functions are implemented as in this example, with error checking of course.

Getpwent() calls **setpwent()** if this has not yet been done. **Setpwent()** first calls **endpwent()** if the password file is still open. (Other implementations may simply rewind the file.)

FILES

/etc/passwd The password file database.

SEE ALSO

cuserid(3), **getlogin**(3), **getgrent**(3), **passwd**(5).

DIAGNOSTICS

Setpwent() has the same return value and error codes as the **open**(2) call it uses to open the password file. The **get**_xxx_**()** functions return NULL on end of file, entry not found, or error. You can set **errno** to zero before the call and check it after.

NOTES

All **get**_xxx_**()** routines return a pointer to static storage that is overwritten in each call.

Only **getpwnam()** and **getpwuid()** are defined by POSIX. The **_MINIX_SOURCE** macro must be defined before including <pwd.h> to make the other functions visible. The **pw_passwd** and **pw_gecos** fields are also not defined by POSIX, but are always visible. Portable code cannot reliably detect errors by setting **errno** to zero. Under Minix it is better to make a **getpwent()** scan if you need to look up several user-id's or names, but portable code had better use several **getpwuid()** or **getpwnam()** calls. The **getpwent()** is usually available on other systems, but may be very expensive.

AUTHOR

Kees J. Bot (kjb@cs.vu.nl)