pwd — Unix Password Database

Purpose: Read user data from Unix password database.

The pwd module can be used to read user information from the Unix password database (usually `/etc/passwd`). The read-only interface returns tuple-like objects with named attributes for the standard fields of a password record.

Index	Attribute	Meaning
0	pw_name	The user’s login name
1	pw_passwd	Encrypted password (optional)
2	pw_uid	User id (integer)
3	pw_gid	Group id (integer)
4	pw_gecos	Comment/full name
5	pw_dir	Home directory
6	pw_shell	Application started on login, usually a command interpreter

Querying All Users

This example prints a report of all of the “real” users on a system, including their home directories (where “real” is defined as having a name not starting with “_”). To load the entire password database, use `getpwall()`. The return value is a list with an undefined order, so it needs to be sorted before the report is printed.

```
# pwd_getpwall.py

import pwd
import operator

# Load all of the user data, sorted by username
all_user_data = pwd.getpwall()
interesting_users = sorted(
    (u for u in all_user_data
     if not u.pw_name.startswith('_')),
    key=operator.attrgetter('pw_name')
)

# Find the longest lengths for a few fields
username_length = max(len(u.pw_name)
                      for u in interesting_users) + 1
home_length = max(len(u.pw_dir)
                  for u in interesting_users) + 1
uid_length = max(len(str(u.pw_uid))
                 for u in interesting_users) + 1

# Print report headers
fmt = ' '.join(['{:<{username_length}}',
                '{:>{uid_length}}',
                '{:<{home_length}}',
                '{}'])
print(fmt.format('User',
                 'UID',
                 'Home Dir',
                 'Description',
                 username_length=username_length,
                 uid_length=uid_length,
                 home_length=home_length))
print('-' * username_length,
      '-' * uid_length,
      '-' * home_length,
      '-' * 20)
```

```
# Print the data
for u in interesting_users:
    print(fmt.format(u.pw_name,
                     u.pw_uid,
                     u.pw_dir,
                     u.pw_gecos,
                     username_length=username_length,
                     uid_length=uid_length,
                     home_length=home_length))
```

Most of the example code above deals with formatting the results nicely. The `for` loop at the end shows how to access fields from the records by name.

```
$ python3 pwd_getpwall.py

User                UID Home Dir          Description
---------- ----------- ----------------- --------------------
Guest               201 /Users/Guest      Guest User
daemon                1 /var/root         System Services
daemon                1 /var/root         System Services
dhellmann           501 /Users/dhellmann  Doug Hellmann
nobody       4294967294 /var/empty        Unprivileged User
nobody       4294967294 /var/empty        Unprivileged User
root                  0 /var/root         System Administrator
root                  0 /var/root         System Administrator
```

Querying User By Name

To read information about one user it is not necessary to read the entire password database. Use `getpwnam()`, to retrieve the information about a user by name.

```
# pwd_getpwnam.py

import pwd
import sys

username = sys.argv[1]
user_info = pwd.getpwnam(username)

print('Username:', user_info.pw_name)
print('Password:', user_info.pw_passwd)
print('Comment :', user_info.pw_gecos)
print('UID/GID :', user_info.pw_uid, '/', user_info.pw_gid)
print('Home    :', user_info.pw_dir)
print('Shell   :', user_info.pw_shell)
```

The passwords on the system where this example was run are stored outside of the main user database in a shadow file, so the password field, when set, is reported as all *.

```
$ python3 pwd_getpwnam.py dhellmann

Username: dhellmann
Password: ********
Comment : Doug Hellmann
UID/GID : 501 / 20
Home    : /Users/dhellmann
Shell   : /bin/bash

$ python3 pwd_getpwnam.py nobody

Username: nobody
Password: *
Comment : Unprivileged User
UID/GID : 4294967294 / 4294967294
Home    : /var/empty
Shell   : /usr/bin/false
```

Querying User By UID

Querying User By UID

It is also possible to look up a user by their numerical user id. This is useful to find the owner of a file:

```
# pwd_getpwuid_fileowner.py
```

```
import pwd
import os

filename = 'pwd_getpwuid_fileowner.py'
stat_info = os.stat(filename)
owner = pwd.getpwuid(stat_info.st_uid).pw_name

print('{} is owned by {} ({})'.format(
    filename, owner, stat_info.st_uid))
```

```
$ python3 pwd_getpwuid_fileowner.py

pwd_getpwuid_fileowner.py is owned by dhellmann (501)
```

The numeric user id is can also be used to find information about the user currently running a process:

```
# pwd_getpwuid_process.py
```

```
import pwd
import os

uid = os.getuid()
user_info = pwd.getpwuid(uid)
print('Currently running with UID={} username={}'.format(
    uid, user_info.pw_name))
```

```
$ python3 pwd_getpwuid_process.py

Currently running with UID=501 username=dhellmann
```

See also

- Standard library documentation for pwd
- spwd – Secure password database access for systems using shadow passwords.
- grp – The grp module reads Unix group information.

Unix-specific Services

grp — Unix Group Database

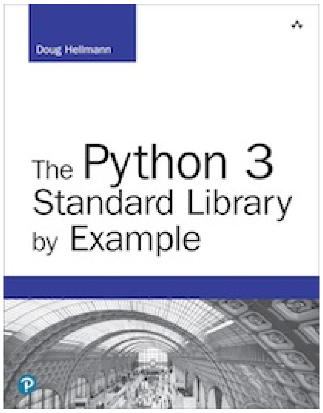
Quick Links

Querying All Users
Querying User By Name
Querying User By UID

This page was last updated 2016-12-03.

Navigation

Unix-specific Services
grp — Unix Group Database



The output from all the example programs from PyMOTW-3 has been generated with Python 3.7.1, unless otherwise noted. Some of the features described here may not be available in earlier versions of Python.

Looking for examples for Python 2?

This Site

Module Index
Index

Other Writing

Blog
The Python Standard Library By Example