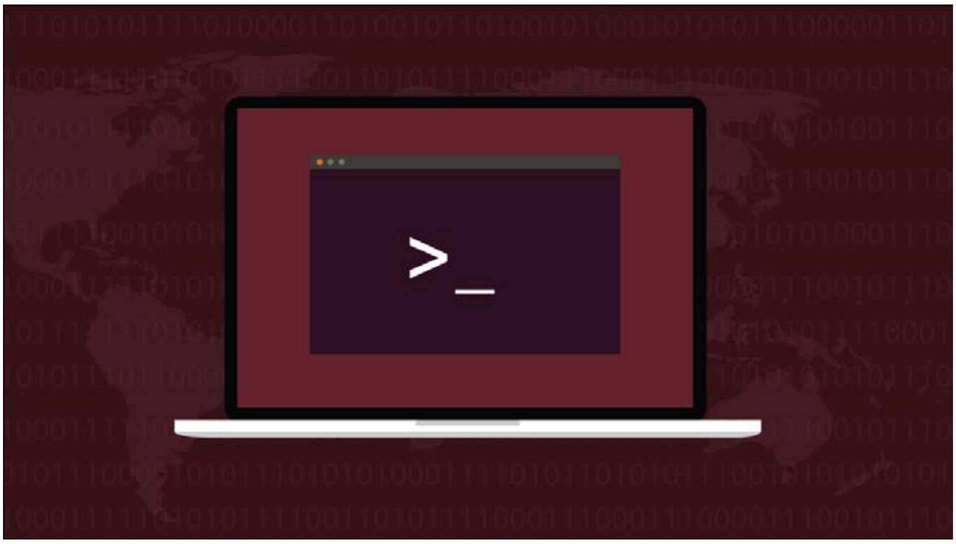
How-To Geek

Linux

How to List Users in Linux

DAVE MCKAY @thegurkha
MAY 18, 2022, 9:00 AM EDT | 6 MIN READ



fatmawati achmad zaenuri/Shutterstock.com

Linux is a multiuser operating system, so creating numerous user accounts is easy. Over time, it's easy to lose track of which accounts are required. Listing user accounts helps you manage them.

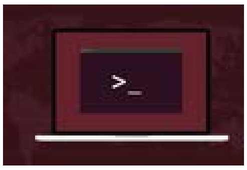
User Accounts

Advances in technology often bring their own new problems. As soon as computers were able to support multiple users, the need to ring-fence and encapsulate each person's work from everyone else became apparent. This led to the concept of user accounts. Each user has a named ID and a password. These are the credentials that let them log into their account. Their files are kept in an area that is private to each user.

On a busy system, it is easy to lose sight of which accounts you have created, and which are no longer needed. From a security point of view, it is bad practice to keep user accounts that you no longer need to be configured and accessible on your computer. You should remove those users.

Even if you don't have other people using your computer you might have created some accounts just to learn how to do it, or to learn and practice administration processes.

The first step is to list the user accounts that are configured on your computer. That lets you review them and make a judgment call on which can be deleted. There are several methods to list users. No matter which distribution you're using, these techniques should work for you without needing to install any applications or utilities.

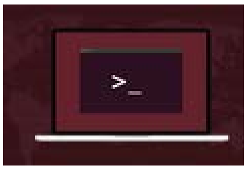


RELATED
[How to Delete a User on Linux (and Remove Every Trace)]

List Users With the cat Command

A list of the configured users is maintained, along with information about each user, in the "/etc/passwd" file. This is a text file that regular users can list to the terminal window. You don't need to use `sudo` to look into the "/etc/passwd" file.

We can use the `cat` command to send the contents of the "/etc/passwd" file to the terminal window. This will list the entire contents of the file. This means you'll also see the entries for user accounts that are owned by processes and the system, not by people.

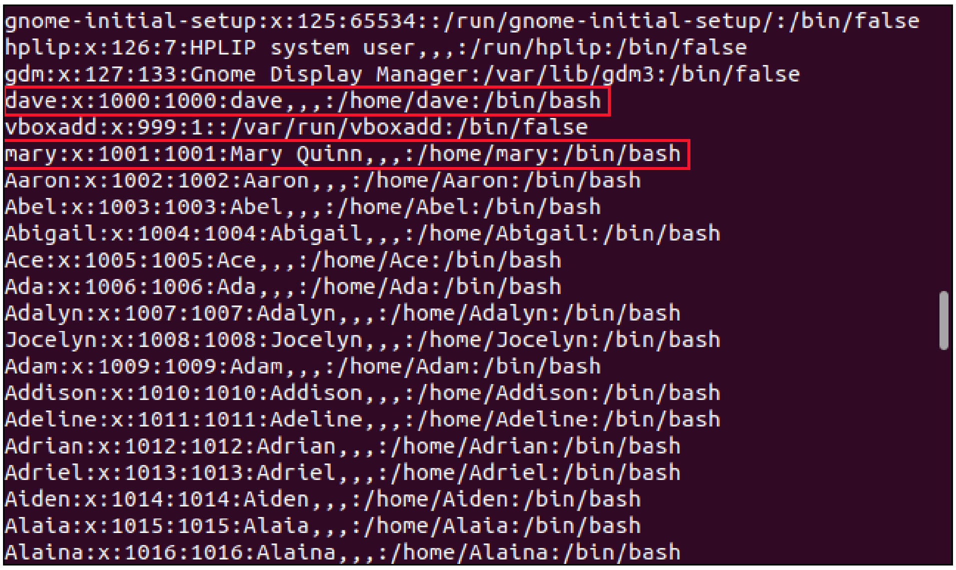


RELATED
[How to Change User Data With chfn and usermod on Linux]

```
cat /etc/passwd
```

```
dave@Ubuntu-22-04:~$ cat /etc/passwd
```

There's a line of dense information reported for each user account.



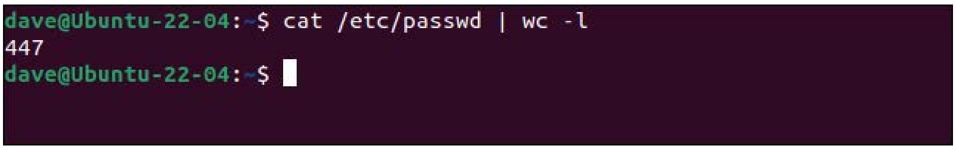
```
gnome-initial-setup:x:125:65534::/run/gnome-initial-setup/:/bin/false
hplip:x:126:7:HPLIP system user,,,:/run/hplip:/bin/false
gdm:x:127:133:Gnome Display Manager:/var/lib/gdm3:/bin/false
dave:x:1000:1000:dave,,,:/home/dave:/bin/bash
vboxadd:x:999:1::/var/run/vboxadd:/bin/false
mary:x:1001:1001:Mary Quinn,,,:/home/mary:/bin/bash
Aaron:x:1002:1002:Aaron,,,:/home/Aaron:/bin/bash
Abel:x:1003:1003:Abel,,,:/home/Abel:/bin/bash
Abigail:x:1004:1004:Abigail,,,:/home/Abigail:/bin/bash
Ace:x:1005:1005:Ace,,,:/home/Ace:/bin/bash
Ada:x:1006:1006:Ada,,,:/home/Ada:/bin/bash
Adalyn:x:1007:1007:Adalyn,,,:/home/Adalyn:/bin/bash
Jocelyn:x:1008:1008:Jocelyn,,,:/home/Jocelyn:/bin/bash
Adam:x:1009:1009:Adam,,,:/home/Adam:/bin/bash
Addison:x:1010:1010:Addison,,,:/home/Addison:/bin/bash
Adeline:x:1011:1011:Adeline,,,:/home/Adeline:/bin/bash
Adrian:x:1012:1012:Adrian,,,:/home/Adrian:/bin/bash
Adriel:x:1013:1013:Adriel,,,:/home/Adriel:/bin/bash
Aiden:x:1014:1014:Aiden,,,:/home/Aiden:/bin/bash
Alaia:x:1015:1015:Alaia,,,:/home/Alaia:/bin/bash
Alaina:x:1016:1016:Alaina,,,:/home/Alaina:/bin/bash
```

The information for the user account called "dave" contains these pieces of information, with colons "`:`" between them.

- **dave**: The name of the user account. Usually the name of the person who owns the account.
- **x**: At one time, this held the password for the account. Nowadays, passwords are stored in the "/etc/shadow" file. The "x" means the password is in that file.
- **1000**: The user ID for this account. All user accounts have a unique numeric ID. Regular user accounts usually start at 1000, with each new account taking the next free ID, such as 1001, 1002, and so on.
- **1000**: The group ID of the default group the user belongs to. In normal circumstances, the default group has the same value as the user ID.
- **dave,,,**: A collection of optional extra information about the user. This field contains data with commas "`,`" between them. They can hold things like the full name of the user, their office number, and their telephone number. The entry for user account "mary" shows her full name is Mary Quinn.
- **/home/dave**: The path to the user's home folder.
- **/bin/bash**: The default shell for this user.

If we pipe the output from this command through the `wc` utility and use the `-l` (lines) option we can count the lines in the file. That'll give us the number of accounts configured on this computer.

```
cat /etc/passwd | wc -l
```



```
dave@Ubuntu-22-04:~$ cat /etc/passwd | wc -l
447
dave@Ubuntu-22-04:~$
```

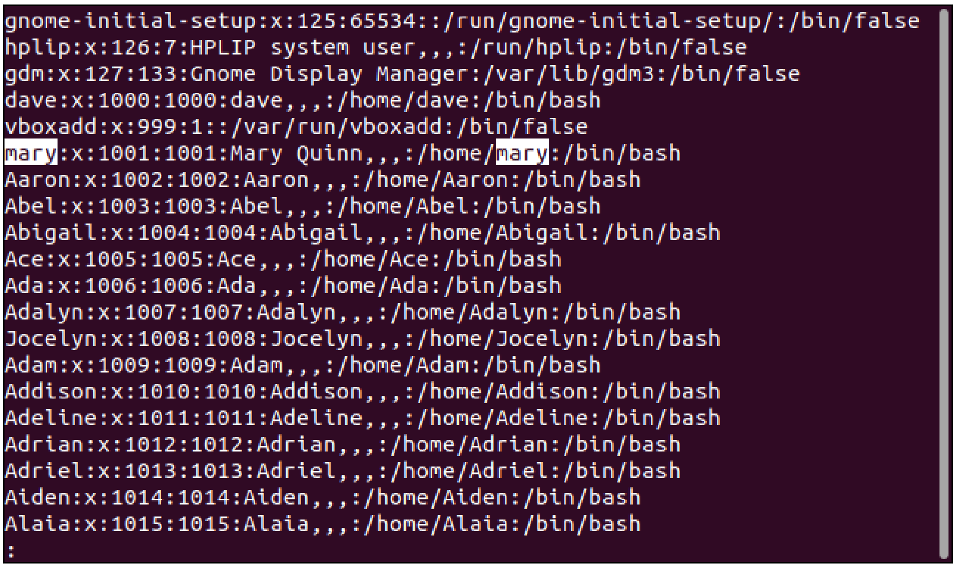
That figure includes the system accounts and users created by applications. There are about 400 regular users configured on this computer. Your result is likely to be a lot less.

With that many accounts, it's more convenient to use `less` to view the "/etc/passwd" file.

```
less /etc/passwd
```

```
dave@Ubuntu-22-04:~$ less /etc/passwd
```

Using `less` also allows you to search within the output, should you want to look for a particular user account.



```
gnome-initial-setup:x:125:65534::/run/gnome-initial-setup/:/bin/false
hplip:x:126:7:HPLIP system user,,,:/run/hplip:/bin/false
gdm:x:127:133:Gnome Display Manager:/var/lib/gdm3:/bin/false
dave:x:1000:1000:dave,,,:/home/dave:/bin/bash
vboxadd:x:999:1::/var/run/vboxadd:/bin/false
mary:x:1001:1001:Mary Quinn,,,:/home/mary:/bin/bash
Aaron:x:1002:1002:Aaron,,,:/home/Aaron:/bin/bash
Abel:x:1003:1003:Abel,,,:/home/Abel:/bin/bash
Abigail:x:1004:1004:Abigail,,,:/home/Abigail:/bin/bash
Ace:x:1005:1005:Ace,,,:/home/Ace:/bin/bash
Ada:x:1006:1006:Ada,,,:/home/Ada:/bin/bash
Adalyn:x:1007:1007:Adalyn,,,:/home/Adalyn:/bin/bash
Jocelyn:x:1008:1008:Jocelyn,,,:/home/Jocelyn:/bin/bash
Adam:x:1009:1009:Adam,,,:/home/Adam:/bin/bash
Addison:x:1010:1010:Addison,,,:/home/Addison:/bin/bash
Adeline:x:1011:1011:Adeline,,,:/home/Adeline:/bin/bash
Adrian:x:1012:1012:Adrian,,,:/home/Adrian:/bin/bash
Adriel:x:1013:1013:Adriel,,,:/home/Adriel:/bin/bash
Aiden:x:1014:1014:Aiden,,,:/home/Aiden:/bin/bash
Alaia:x:1015:1015:Alaia,,,:/home/Alaia:/bin/bash
:
```

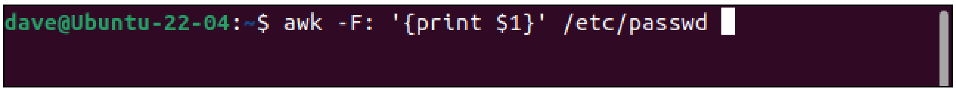
The awk Command

Using [the awk command](#) we can display just the username. This can be useful when you're writing a script that needs to do something to a lot of user accounts. Listing the user account names and redirecting them into a text file can be a great time saver. All you need to do then is copy and paste the rest of the command onto each line.

We'll tell awk to use the colon ":" as the field separator, and to print the first field. We'll use the -F (field separator) option.

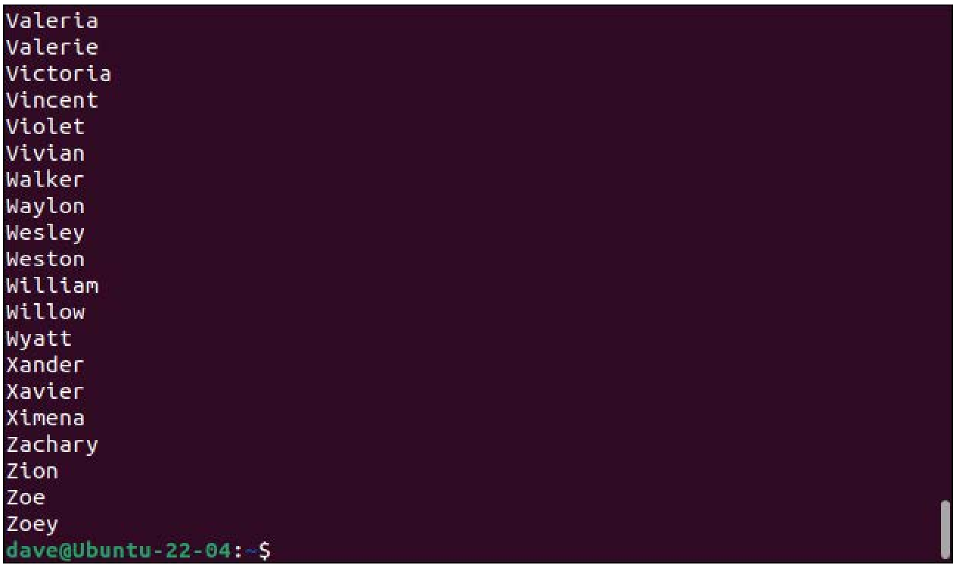
RELATED
How to Use the awk Command on Linux

```
awk -F: '{print $1}' /etc/passwd
```



```
dave@Ubuntu-22-04:~$ awk -F: '{print $1}' /etc/passwd
```

The user account names are written to the terminal window without any of the other account information.

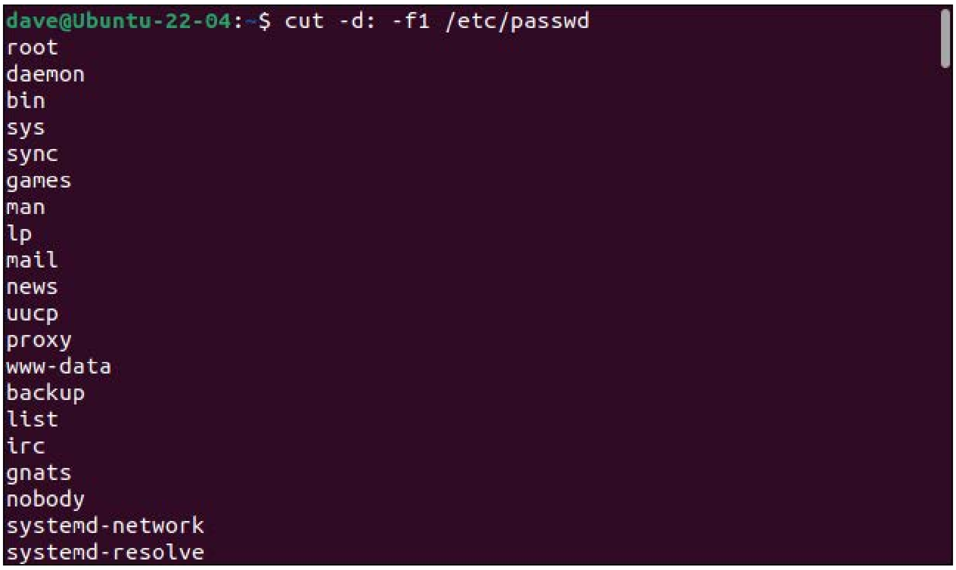


```
Valeria
Valerie
Victoria
Vincent
Violet
Vivian
Walker
Waylon
Wesley
Weston
William
Willow
Wyatt
Xander
Xavier
Ximena
Zachary
Zion
Zoe
Zoey
dave@Ubuntu-22-04:~$
```

The cut Command

We can achieve the same sort of thing using the `cut` command. We need to use the `-d` (delimiter) option and ask it to select the first field only, using the `-f` (fields) option.

```
cutr -d: -f1
```



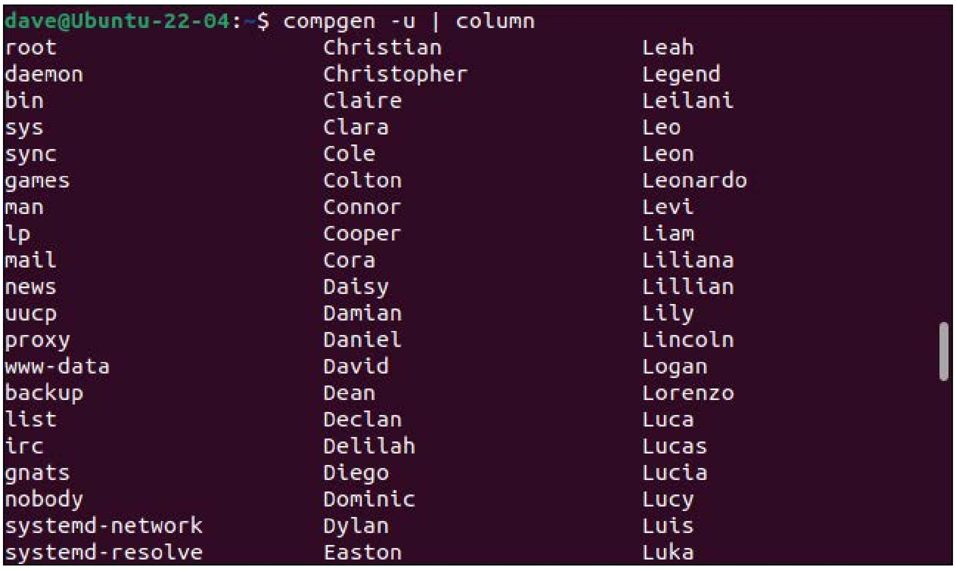
```
dave@Ubuntu-22-04:~$ cut -d: -f1 /etc/passwd
root
daemon
bin
sys
sync
games
man
lp
mail
news
uucp
proxy
www-data
backup
list
irc
gnats
nobody
systemd-network
systemd-resolve
```

This lists all of the user accounts, including the system and other non-human accounts.

The compgen Command

The `compgen` command can be used with the `-u` (user) option to list the user accounts. We'll pipe the output through the `column` command to list the user accounts in columns, instead of one long list with a single user name per line.

```
compgen -u | column
```



```
dave@Ubuntu-22-04:~$ compgen -u | column
root                    Christian               Leah
daemon                  Christopher             Legend
bin                     Claire                  Leilani
sys                     Clara                   Leo
sync                    Cole                    Leon
games                   Colton                  Leonardo
man                     Connor                  Levi
lp                      Cooper                  Liam
mail                    Cora                    Liliana
news                    Daisy                   Lillian
uucp                    Damian                  Lily
proxy                   Daniel                  Lincoln
www-data                David                   Logan
backup                  Dean                    Lorenzo
list                    Declan                  Luca
irc                     Delilah                 Lucas
gnats                   Diego                   Lucia
nobody                  Dominic                 Lucy
systemd-network         Dylan                   Luis
systemd-resolve         Easton                  Luka
```

Again, the first user accounts listed belong to processes, not humans.

UID MIN and UID MAX

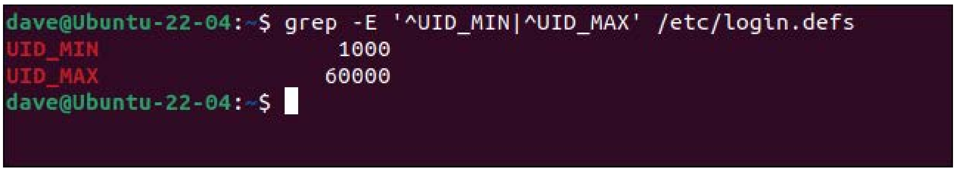
User accounts are given a numeric ID, which we saw earlier. Usually, the regular human user accounts start at 1000, and the system, non-human, user accounts start at 0. The ID of the root account is 0.

If we can verify the lowest and highest possible user IDs, we can use that information to select the user accounts that are between those two values. That will let us select only the user accounts belonging to real people.

Linux keeps track of these two values using configuration parameters called `UID_MIN` and `UID_MAX` . These are held in the "/etc/login.defs" file. We can easily see these values using `grep`.

We're going to use the `-E` (extended regex) option. Our search string looks for lines that begin with "UID_MIN" or "UID_MAX" in the "/etc/login.defs" file. The caret "^" represents the beginning of a line.

```
grep -E '^UID_MIN|^UID_MAX' /etc/login.defs
```



```
dave@Ubuntu-22-04:~$ grep -E '^UID_MIN|^UID_MAX' /etc/login.defs
UID_MIN                  1000
UID_MAX                 60000
dave@Ubuntu-22-04:~$ 
```

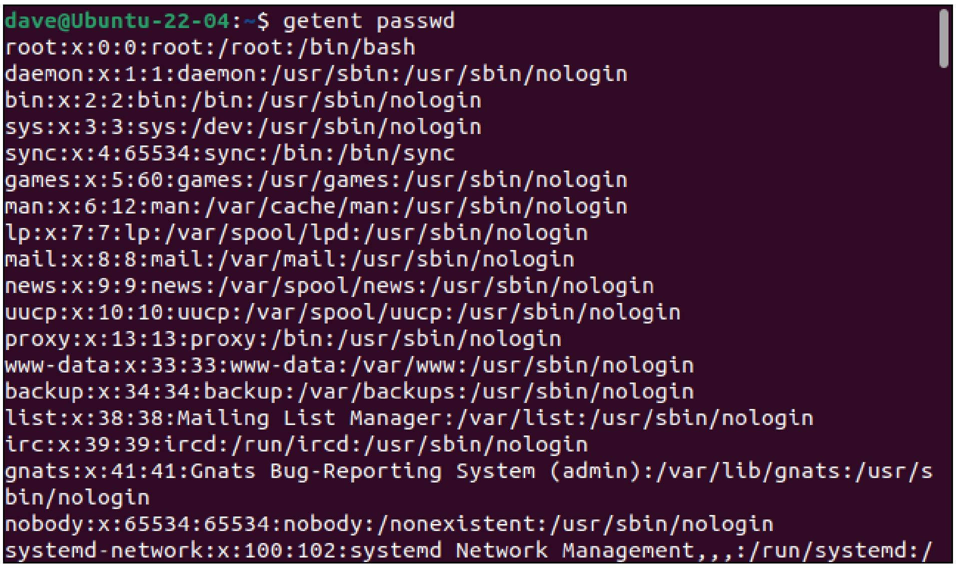
The range for user IDs on this computer is from 1000 to 60,000.

RELATED: ***How to Use Regular Expressions (regexes) on Linux***

The getent Command

The `getent` command reads information from system databases. We can tell it to list the entries in the "/etc/passwd" file by using "passwd" as a parameter.

```
getent passwd
```



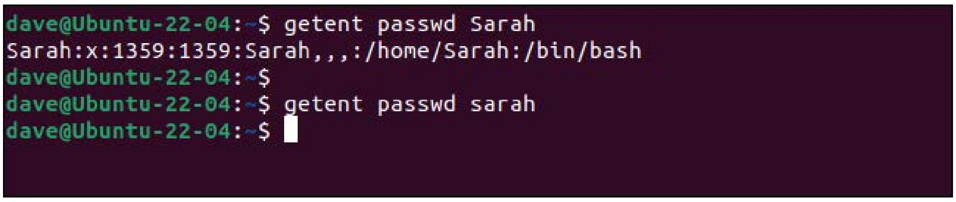
```
dave@Ubuntu-22-04:~$ getent passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/s
bin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/
```

This gives us the same readout we can get using `cat`. But where `getent` shines is by accepting values known as "keys." A key dictates which information `getent` reports on. If we want to see the entry for a single user, we can pass in their user account name on the command line.

```
getent passwd Sarah
```

Note that the user account name is case-sensitive.

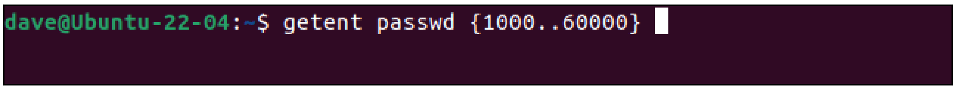
```
getent passwd sarah
```



```
dave@Ubuntu-22-04:~$ getent passwd Sarah
Sarah:x:1359:1359:Sarah,,,:/home/Sarah:/bin/bash
dave@Ubuntu-22-04:~$
dave@Ubuntu-22-04:~$ getent passwd sarah
dave@Ubuntu-22-04:~$
```

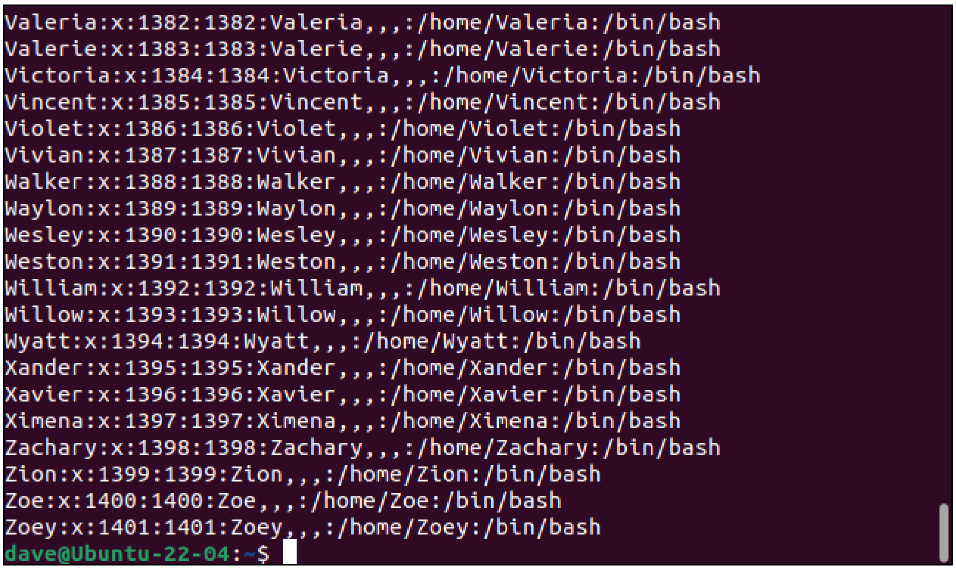
We can also pass in the upper and lower limits of the user account IDs we want to see. To see absolutely all the regular user accounts, we can use the values from `UID_MIN` and `UID_MAX`.

```
getent passwd {1000..60000}
```



```
dave@Ubuntu-22-04:~$ getent passwd {1000..60000}
```

This takes some time to run. Eventually, you'll be returned to the command prompt.

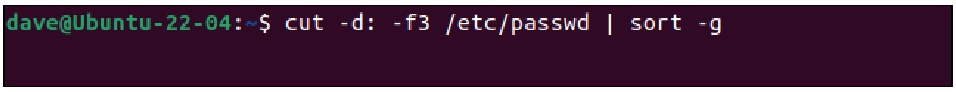


```
Valeria:x:1382:1382:Valeria,,,:/home/Valeria:/bin/bash
Valerie:x:1383:1383:Valerie,,,:/home/Valerie:/bin/bash
Victoria:x:1384:1384:Victoria,,,:/home/Victoria:/bin/bash
Vincent:x:1385:1385:Vincent,,,:/home/Vincent:/bin/bash
Violet:x:1386:1386:Violet,,,:/home/Violet:/bin/bash
Vivian:x:1387:1387:Vivian,,,:/home/Vivian:/bin/bash
Walker:x:1388:1388:Walker,,,:/home/Walker:/bin/bash
Waylon:x:1389:1389:Waylon,,,:/home/Waylon:/bin/bash
Wesley:x:1390:1390:Wesley,,,:/home/Wesley:/bin/bash
Weston:x:1391:1391:Weston,,,:/home/Weston:/bin/bash
William:x:1392:1392:William,,,:/home/William:/bin/bash
Willow:x:1393:1393:Willow,,,:/home/Willow:/bin/bash
Wyatt:x:1394:1394:Wyatt,,,:/home/Wyatt:/bin/bash
Xander:x:1395:1395:Xander,,,:/home/Xander:/bin/bash
Xavier:x:1396:1396:Xavier,,,:/home/Xavier:/bin/bash
Ximena:x:1397:1397:Ximena,,,:/home/Ximena:/bin/bash
Zachary:x:1398:1398:Zachary,,,:/home/Zachary:/bin/bash
Zion:x:1399:1399:Zion,,,:/home/Zion:/bin/bash
Zoe:x:1400:1400:Zoe,,,:/home/Zoe:/bin/bash
Zoey:x:1401:1401:Zoey,,,:/home/Zoey:/bin/bash
dave@Ubuntu-22-04:~$
```

The reason for the long execution time is that `getent` tries to find matches for all of the user account values right up to 60000.

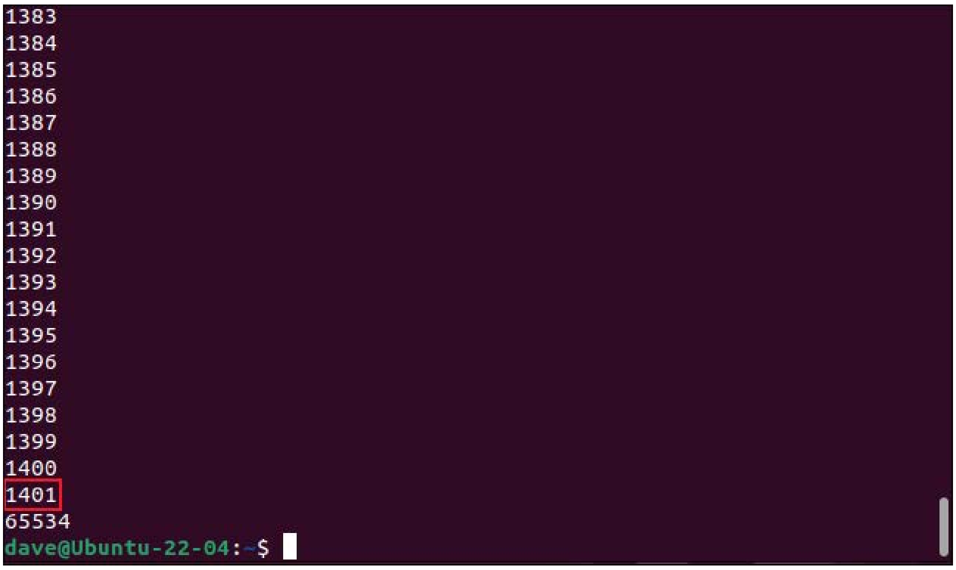
Let's see what the highest user account ID is. We'll use the `cut` command, but this time we'll ask for field three, the user ID field. We'll pipe the output through `sort` and use the `-g` (general numeric sort) option.

```
cut -d: -f3 /etc/passwd | sort -g
```



```
dave@Ubuntu-22-04:~$ cut -d: -f3 /etc/passwd | sort -g
```

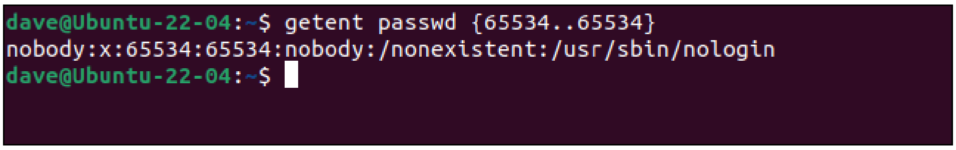
The highest ID value of a human-owned user account is 1401.



```
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
65534
dave@Ubuntu-22-04:~$
```

User id 65534 is assigned to the system concept of "nobody."

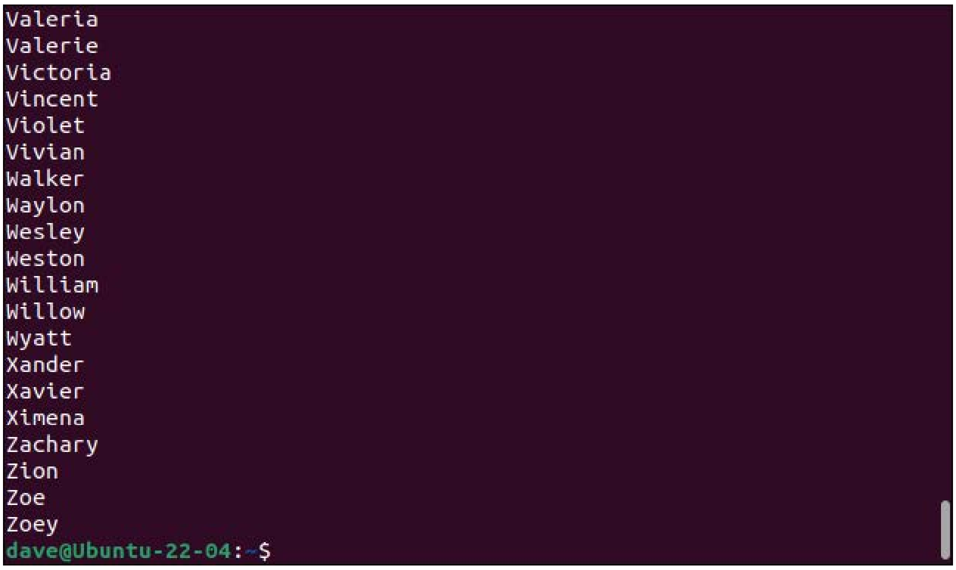
```
getent passwd {65534..65534}
```



```
dave@Ubuntu-22-04:~$ getent passwd {65534..65534}
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
dave@Ubuntu-22-04:~$
```

So we know that instead of using the `UID_MAX` value of 60000, on this computer we can use a more realistic value like 1500. That'll speed things up nicely. We'll also pipe the output through `cut` to extract just the names of the user accounts.

```
getent passwd {1000..1500} | cut -d: -f1
```

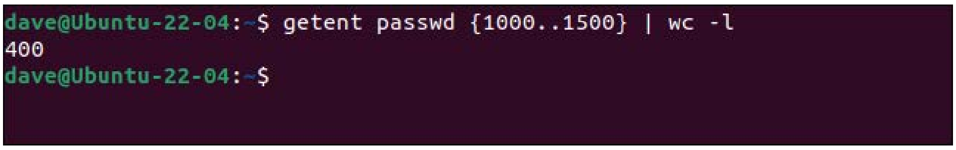


```
Valeria
Valerie
Victoria
Vincent
Violet
Vivian
Walker
Waylon
Wesley
Weston
William
Willow
Wyatt
Xander
Xavier
Ximena
Zachary
Zion
Zoe
Zoey
dave@Ubuntu-22-04:~$
```

The users are listed and we're returned immediately to the command prompt.

Instead of piping the output through `cut`, let's pipe the output through `wc` and count the lines once more. That'll give us the number of "real" user accounts.

```
getent passwd {1000..1500} | wc -l
```



```
dave@Ubuntu-22-04:~$ getent passwd {1000..1500} | wc -l
400
dave@Ubuntu-22-04:~$
```

We can now see that on this computer, definitively, there are 400 configured, human-owned, user accounts.

Power and Simplicity

One of these techniques is sure to suit your needs when you need to review the user accounts on a Linux computer. These commands should be present on all distributions, and none of them require sudo access, so they are all available to every user.

RELATED: ***How to Control sudo Access on Linux***

DAVE MCKAY

Dave McKay first used computers when punched paper tape was in vogue, and he has been programming ever since. After over 30 years in the IT industry, he is now a full-time technology journalist. During his career, he has worked as a freelance programmer, manager of an international software development team, an IT services project manager, and, most recently, as a Data Protection Officer. His writing has been published by howtogeek.com, cloudsavvyit.com, itenterpriser.com, and opensource.com. Dave is a Linux evangelist and open source advocate. **READ FULL BIO »**