A sample output is as follows:

```
# ./user_adm.sh -details test
Login: test                 Name:
Directory: /home/test                 Shell: /bin/sh
Last login Tue Dec 21 00:07 (IST) on pts/1 from localhost
No mail.
No Plan.
Last password change          : Dec 20, 2010
Password expires          : never
Password inactive         : never
Account expires             : Oct 10, 2010
Minimum number of days between password change    : 0
Maximum number of days between password change    : 99999
Number of days of warning before password expires  : 7
```

How it works...

The `user_adm.sh` script can be used to perform many user management tasks. You can follow the `usage()` text for the proper usage of the script which is called when any of the parameters given by the user is wrong or has the `-usage` parameter. A case statement is used to match the command arguments and execute the corresponding commands according to that. The valid command options for the `user_adm.sh` script are: `-adduser`, `-deluser`, `-shell`, `-disable`, `-enable`, `-expiry`, `-passwd`, `-newgroup`, `-delgroup`, `-addgroup`, `-details`, and `-usage`. When the `*)` case is matched, it means it is a wrong option and hence, `usage()` is invoked. For each match case, we have used `[ $# -ne 3 ] &&` usage. It is used for checking the number of arguments. If the number of command arguments are not equal to the required number, the `usage()` function is invoked and the script will exit without executing further. In order to run the user management commands, the script needs to be run as root and hence, a check for the user ID 0 (root has user ID 0) is performed.

Let's explain each case one by one:

- `-useradd`: The `useradd` command can be used to create a new user. It has the following syntax:

  ```
  useradd USER -p PASSWORD
  ```

- The `-m` option is used to create the home directory. It is also possible to provide the full name of the user by using the `-c FULLNAME` option.
- `-deluser`: The `deluser` command can be used to remove the user. The syntax is as follows:

  ```
  deluser USER
  ```

- `--remove-all-files` is used to remove all files associated with the user including the home directory.
- `-shell`: The `chsh` command is used to change the default shell for the user. The syntax is:

  ```
  chsh USER -s SHELL
  ```

- `-disable` and `-enable`: The `usermod` command is used to manipulate several attributes related to user accounts. `usermod -L USER` locks the user account and `usermod -U USER` unlocks the user account.
- `-expiry`: The `chage` command is used to manipulate user account expiry information. The syntax is:

  ```
  chage -E DATE
  ```

  There are additional options as follows:

  - `-m MIN_DAYS` (set the minimum number of days between password changes to `MIN_DAYS`)
  - `-M MAX_DAYS` (set the maximum number of days during which a password is valid)
  - `-W WARN_DAYS` (set the number of days of warning before a password change is required)

- `-passwd`: The `passwd` command is used to change passwords for the users. The syntax is:

  ```
  passwd USER
  ```

  The command will prompt to enter a new password.

- `-newgroup` and `addgroup`: The `addgroup` command will add a new user group to the system. The syntax is:

  ```
  addgroup GROUP
  ```

  In order to add an existing user to a group use:

  ```
  addgroup USER GROUP
  -delgroup
  ```

  The `delgroup` command will remove a user group. The syntax is:

  ```
  delgroup GROUP
  ```

- `-details`: The `finger USER` command will display the user information for the user which includes details such as user home directory path, last login time, default shell, and so on. The `chage -l` command will display the user account expiry information.