

# Switching Users and Running Commands as Others

## su

One way to start a session as another user on the system is to use the `su` command. If no arguments are supplied to `su`, it assumes you are trying to become the superuser. Executing `su` is the same as executing `su root`. Your current environment is passed to the new shell unless you specify a hyphen (-). In that case, `su` creates an environment like you would expect to see had you logged in as that user.

`su [username]` - Change user ID or become superuser

Common `su` options:

- A hyphen is used to provide an environment similar to what the user would expect had the user logged in directly.
- c command - Specify a command to be executed. If the command is more than one word in length, it needs to be quoted.

```
bob@linuxsvr:~$ export TEST=1
bob@linuxsvr:~$ su oracle
Password:
oracle@linuxsvr:/home/bob$ echo $TEST
1
oracle@linuxsvr:/home/bob$ pwd
/home/bob
oracle@linuxsvr:/home/bob$ exit
exit
bob@linuxsvr:~$ su - oracle
Password:
oracle@linuxsvr:~$ echo $TEST

oracle@linuxsvr:~$ pwd
/home/oracle
oracle@linuxsvr:~$ exit
bob@linuxsvr:~$ su -c 'echo $ORACLE_HOME' oracle
Password:

bob@linuxsvr:~$ su -c 'echo $ORACLE_HOME' - oracle
Password:
/u01/app/oracle/product/current
bob@linuxsvr:~$
```

If you want to know what user you are working as, run the `whoami` command.

`whoami` - Displays the effective username.

```
$ whoami
bob
$ su oracle
Password:
$ whoami
oracle
$
```

## Sudo - Super User Do

Another way to switch users or execute commands as others is to use the `sudo` command. Sudo

allows you to run programs with the security privileges of another user. Like `su`, if no username is specified it assumes you are trying to run commands as the superuser. This is why `sudo` is referred to as super user do. It is commonly used to install, start, and stop applications that require superuser privileges.

`sudo` - Execute a command as another user, typically the superuser.

One advantage of using `sudo` over the `su` command is that you do not need to know the password of the other user. This can eliminate the issues that arise from using shared passwords and generic accounts. When you execute the `sudo` command you are prompted for your password. If the `sudo` configuration permits access, the command is executed. The `sudo` configuration is typically controlled by the system administrator and requires root access to change.

## Using Sudo

Here are the common ways to use the `sudo` command.

`sudo -l` - List available commands.

`sudo command` - Run command as the superuser.

`sudo -u root command` - Same as `sudo command`.

`sudo -u user command` - Run command as user.

`sudo su` - Switch to the superuser account.

`sudo su -` - Switch to the superuser account with an environment like you would expect to see had you logged in as that user.

`sudo su - username` - Switch to the username account with an environment like you would expect to see had you logged in as that user.

```
$ sudo -l
User bob may run the following commands on this host:
(root) NOPASSWD: /etc/init.d/apache2
(fred) NOPASSWD: /opt/fredApp/bin/start
(fred) NOPASSWD: /opt/fredApp/bin/stop
(root) /bin/su - oracle
$ sudo /etc/init.d/apache2 start
* Starting web server apache2
$ sudo -u fred /opt/fredApp/bin/start
Fred's app started as user fred.
$ sudo su - oracle
[sudo] password for bob:
oracle@linuxsvr:~$ whoami
oracle
oracle@linuxsvr:~$ exit
$ whoami
bob
$
```

The output of `sudo -l` displays what commands can be executed with `sudo` and under which account. In the above example, `sudo` will not prompt for a password for the commands preceded with `NOPASSWD`. This type of configuration may be required to automate jobs via cron that require escalated privileges.

