

# Maharana Pratap College of Technology



**Session 2017-2021**

**Computer Science**

**3<sup>rd</sup> Year**

**Linux Assignment**

**Submitted to: -**

**Submitted By: -**

**Anurag Shrivastava Sir**

**Alok Saxena**

Q1 Write commands for performing arithmetic operations in Linux/Unix?

```
a=10
b=20

val=`expr $a + $b`
echo "a + b : $val"

val=`expr $a - $b`
echo "a - b : $val"

val=`expr $a \* $b`
echo "a * b : $val"

val=`expr $b / $a`
echo "b / a : $val"

val=`expr $b % $a`
echo "b % a : $val"

if [ $a == $b ]
then
    echo "a is equal to b"
fi

if [ $a != $b ]
then
    echo "a is not equal to b"
fi
```

The above script will produce the following result –

```
a + b : 30
a - b : -10
a * b : 200
b / a : 2
b % a : 0
a is not equal to b
```

Q.2 Create a file called wlcc.txt with some lines and find how many words lines are presented in that file?

```
$ sudo touch wlcc.txt
```

```
$ sudo touch wlcc.docx
```

```
echo Enter the filename
read file
w=`cat $file | wc -w`
c=`cat $file | wc -c`
l=`grep -c "." $file`
echo Number of characters in $file is $c
echo Number of words in $file is $w
echo Number of lines in $file is $l
```

Q.3 append 10 lines to the file wlcc.txt.

```
echo 'linux is a good language' >> filename.txt

echo 'easy to use' >> filename.txt

echo 'simple' >> filename.txt

echo 'consize' >> filename.txt

echo 'file name is wlcc.txt' >> filename.txt

echo 'txt is used for text type file' >> filename.txt

echo 'we can perform arthematic operations in linux' >> filename.txt

echo 'like' >> filename.txt

echo 'addition' >> filename.txt

echo 'Subraction' >> filename.txt

echo 'Multiplication division etc....' >> filename.txt
```

Append command output to end of file:

```
command >> filename.txt
```

Q.4 Given two files each of which contains names of students write a program to find common names of student.

```
$ bash --version
```

GNU bash, version 3.2.51(1)-release

Copyright (C) 2007 Free Software Foundation, Inc.

```
$ cat > abc
```

123

567

132

```
$ cat > def
```

132

777

321

So the files abc and def have one line in common, the one with "132". Using comm on unsorted files:

```
$ comm abc def
```

123

132

567

132

777

321

```
$ comm -12 abc def # No output! The common line is not found
```

```
$
```

The last line produced no output, the common line was not discovered.

Now use comm on sorted files, sorting the files with process substitution:

```
$ comm <( sort abc ) <( sort def )
```

123

132

321

567

777

```
$ comm -12 <( sort abc ) <( sort def )
```

132

Q5.Create program to find an in node number of any desired file?

```
ls -li filename
```

```
$ ls -li /etc/resolv.conf
```

Sample outputs:

```
25766494 -rwxr-xr-x 1 root root 332 May 5 14:14 /etc/resolv.conf
```

```
$ stat fileName-Here
$ stat /etc/passwd
Sample outputs:
```

```
File: `/etc/passwd'

Size: 1644          Blocks: 8          IO Block: 4096   regular file

Device: fe01h/65025d  Inode: 25766495   Links: 1

Access: (0644/-rw-r--r--)  Uid: (   0/   root)   Gid: (   0/   root)

Access: 2012-05-05 16:29:42.000000000 +0530

Modify: 2012-05-05 16:29:20.000000000 +0530

Change: 2012-05-05 16:29:21.000000000 +0530
```

Q.6 Create program in linux to change file permissions?

```
[me@linuxbox me]$
```

```
-rwxr-xr-x 1 root root 316848 Feb 27 2000 /bin/bash
[me@linuxbox me]$ chmod 600 some_file
```

Q7. Execute Shell commands for vi editor?

**Command Mode:** When vi starts up, it is in Command Mode. This mode is where vi interprets any characters we type as commands and thus does not display them in the window. This mode allows us to move through a file, and to delete, copy, or paste a piece of text.

To enter into Command Mode from any other mode, it requires pressing the [Esc] key. If we press [Esc] when we are already in Command Mode, then vi will beep or flash the screen.

**Insert mode:** This mode enables you to insert text into the file. Everything that's typed in this mode is interpreted as input and finally, it is put in the file. The vi always starts in command mode. To enter text, you must be in insert mode. To come in insert mode you simply type i. To get out of insert mode, press the Esc key, which will put you back into command mode.

**Last Line Mode(Escape Mode):** Line Mode is invoked by typing a colon [:], while vi is in Command Mode. The cursor will jump to the last line of the screen and vi

will wait for a command. This mode enables you to perform tasks such as saving files, executing commands.

Starting the vi Editor:

There are following way you can start using vi editor :

Commands and their Description

vi filename: Creates a new file if it already not exist, otherwise opens existing file.

vi -R filename : Opens an existing file in read only mode.

view filename : Opens an existing file in read only mode.

Moving within a File(Navigation):

To move around within a file without affecting text must be in command mode (press Esc twice). Here are some of the commands can be used to move around one character at a time.

Commands and their Description

k : Moves the cursor up one line.

j : Moves the cursor down one line.

h : Moves the cursor to the left one character position.

l : Moves the cursor to the right one character position.

0 or | : Positions cursor at beginning of line.

\$ : Positions cursor at end of line.

W : Positions cursor to the next word.

B : Positions cursor to previous word.

( : Positions cursor to beginning of current sentence.

) : Positions cursor to beginning of next sentence.

H : Move to top of screen.

nH : Moves to nth line from the top of the screen.

M : Move to middle of screen.

L : Move to bottom of screen.

nL : Moves to nth line from the bottom of the screen.

colon along with x : Colon followed by a number would position the cursor on line number represented by x.

Control Commands(Scrolling): There are following useful commands which can be used along with Control Key:

Commands and their Description:

CTRL+d : Move forward 1/2 screen.

CTRL+f : Move forward one full screen.

CTRL+u : Move backward 1/2 screen.

CTRL+b : Move backward one full screen.

CTRL+e : Moves screen up one line.

CTRL+y : Moves screen down one line.

CTRL+u : Moves screen up 1/2 page.

CTRL+d : Moves screen down 1/2 page.

CTRL+b : Moves screen up one page.

CTRL+f : Moves screen down one page.

CTRL+l : Redraws screen.

Editing and inserting in Files(Entering and Replacing Text): To edit the file, we need to be in the insert mode. There are many ways to enter insert mode from the command mode.

I : Inserts text before current cursor location.

O : Inserts text at beginning of current line.

A : Inserts text after current cursor location.

EA : Inserts text at end of current line.

O : Creates a new line for text entry below cursor location.

O : Creates a new line for text entry above cursor location.

R : Replace single character under the cursor with the next character typed.

R : Replaces text from the cursor to right.

S : Replaces single character under the cursor with any number of characters.

S :Replaces entire line.

Deleting Characters: Here is the list of important commands which can be used to delete characters and lines in an opened file.

X : Deletes the character under the cursor location.

X : Deletes the character before the cursor location.

Dw : Deletes from the current cursor location to the next word.

d^ : Deletes from current cursor position to the beginning of the line.

d\$ : Deletes from current cursor position to the end of the line.

Dd : Deletes the line the cursor is on.

Copy and Past Commands: Copy lines or words from one place and paste them on another place by using the following commands.

Yy : Copies the current line.

9yy : Yank current line and 9 lines below.

p : Puts the copied text after the cursor.

P : Puts the yanked text before the cursor.

Save and Exit Commands of the ex Mode : Need to press [Esc] key followed by the colon (:) before typing the following commands:

q : Quit



q! : Quit without saving changes i.e. discard changes.

r fileName : Read data from file called fileName.

wq : Write and quit (save and exit).

w fileName : Write to file called fileName (save as).

w! fileName : Overwrite to file called fileName (save as forcefully).

!cmd : Runs shell commands and returns to Command mode.

Searching and Replacing in (ex Mode): vi also has powerful search and replace capabilities. The formal syntax for searching is:

**:s/string**

Q8. Write shell script to find the smallest number among 3 numbers read through keyboard ?

echo Enter 3 numbers with spaces in between

read a b c

s=\$a

if [ \$b -lt \$s ]

then

s=\$b

fi

if [ \$c -lt \$s ]

then

s=\$c

fi

echo Smallest of \$a \$b \$c is \$s

Q9. Installation of SAMBA,APACHE,TOMCAT?

1 - Create a low privilege user

Running Tomcat as root introduces the unnecessary risk that a compromised Tomcat instance could yield control over your entire server. Thus, creating a user with low privileges to run Tomcat should be standard practice when installing new instances.

Logged in as the root user, run the following command:

```
# useradd -g 46 -s /sbin/nologin -d /opt/tomcat/temp tomcat
```

This command creates a new user named 'tomcat' belonging to the nobody group, with the /sbin/nologin/ shell and a locked password.

## 2 - Download the latest binary release

While there are a number of distribution-specific Tomcat packages available, the quality of these releases varies.

Unless you are using one of the better maintained releases (for example, the Ubuntu/Debian package, which is maintained by MuleSoft's own Jason Brittain, author of O'Reilly's *The Definitive Guide to Apache Tomcat* and lead developer of [Tcat](#)), downloading the latest release directly from Apache is the only way to be sure that you are getting the latest bug and stability fixes.

The trade-off is that the installation may not be as tightly integrated into the system, and may require extra legwork to monitor or update (as opposed to executing a simple package manager command).

The latest stable release can always be found on the Apache Tomcat [download page](#) on Apache's website.

## 3 - Move distribution into place and uncompress

Next, copy Tomcat to the directory you want to install it in, and extract the package:

```
# cp apache-tomcat-6.x.x.tar.gz CurrentDirectory/NewHome
```

```
# cd $NewHome
```

```
# tar zxvf apache-tomcat-6.0.29.tar.gz
```

## 4 - Change permissions

Finally, change the permissions of the folder to allow the user you created to run Tomcat in step one to read and write to the folder (you'll need root access again):

```
# chown -R tomcat apache-tomcat-6.0.29
```

## 5 - Start Tomcat

You should now be able to start Tomcat from the command line by running `startup.sh`, the start script included with Tomcat, or by running `catalina.sh` with the 'start' parameter.

## Platform specific instructions

### Debian and Ubuntu Linux

The standard Linux installation instructions provided above can be used without modification to successfully install Tomcat on Debian or Ubuntu Linux.

However, a number of quirks will affect a manual Debian/Ubuntu installation. For example, Tomcat's STDOUT and STDERR logs will be appended to the main syslog, instead of appearing in Tomcat's default log file, 'catalina.out'.

Also, as the Apache distributions of Tomcat include no Debian/Ubuntu-specific init script, a custom script must be written and installed if Tomcat needs to be run as a service. These are not big problems, but they can be a hassle.

Although it is normally a good idea to avoid re-packaged versions of Tomcat from repositories, the Debian / Ubuntu package (they share the same upstream) is maintained by MuleSoft's own Jason Brittain, author of O'Reilly's Definitive Guide To Apache Tomcat.

Thanks to Jason's work, most users should find that a simple 'apt get' command is now the easiest and best way to install Tomcat on the Debian and Ubuntu platforms.

Jason's package includes a custom init script that makes Tomcat restarts more reliable, and some custom Tomcat configurations that help users to avoid the most common problems with new manual installations, such as disabling the SecurityManager by default and fixing port binding issues.

If you'd like some more information about the package, Jason has written two in-depth blogs about his improvements, which you can read [here](#) and [here](#) on the MuleSoft blog.

### Red Hat Linux

The standard Linux installation instructions provided above can be used without modification to successfully install Tomcat on Red Hat Linux.

Although a Tomcat package is provided in the YUM repositories, it is not frequently updated, and so a manual installation is not recommended.

If you want to run Tomcat as a service, as with Debian and Ubuntu, you will need to write a custom init.d script, similar to the bare bones example below, and place it in Red Hat's "/etc/init.d" folder:

```
#!/bin/bash
```

```
#
```

```
# tomcat
#

# chkconfig:

# description:  Start up the Tomcat servlet engine.

# Source function library.

. /etc/init.d/functions

RETVAL=$?

CATALINA_HOME="/path/to/tomcat"

case "$1" in

    start)

        if [ -f $CATALINA_HOME/bin/startup.sh ];

            then

                echo $"Starting Tomcat"

                /bin/su -s /bin/bash tomcat $CATALINA_HOME/bin/startup.sh

            fi

        ;;

    stop)

        if [ -f $CATALINA_HOME/bin/shutdown.sh ];

            then

                echo $"Stopping Tomcat"

                /bin/su tomcat $CATALINA_HOME/bin/shutdown.sh

            fi

        ;;

    *)
```

```

    echo $"Usage: $0 {start|stop}"

exit 1

;;

esac

exit $RETVAL

```

## Gentoo Linux

The standard Linux installation instructions provided above can be used without modification to successfully install Tomcat on Debian or Ubuntu Linux. However, the installation will not comply with Gentoo / LHS standards.

A distribution of Tomcat that splits Tomcat into multiple Gentoo-friendly pieces is available in the Gentoo repositories. Documentation is available on [gentoo.org](http://gentoo.org). However, the guide has not been updated since 2007, so users should proceed with caution.

To install Tomcat from the Gentoo repositories, use the following command:

```
emerge tomcat
```

The Gentoo distribution includes a custom init script. Once you have installed Tomcat, it can be started with the following command:

```
# /etc/init.d/tomcat-6 start
```

To enable Tomcat as a service, so that it automatically starts at boot time, use the following command:

```
rc-update add tomcat-6 default
```

## Q 10.Implementation of DNS,LADAP services?

### Installing OpenLDAP

To install OpenLDAP you have to install openldap, openldap-servers and openldap-clients packages.

```
$ yum -y install openldap openldap-servers openldap-clients
```

Or if you are using CentOS 7, you can use dnf or Dandified Yum

```
$ dnf -y install openldap openldap-servers openldap-clients
```

If you are using a Debian based system like Ubuntu, you can install it like this:



```
1 $ sudo apt-get install slapd
2
3 $ sudo apt-get install ldap-utils
```

Then we can enable the service to run automatically at startup:

```
$ systemctl enable slapd
```

## Configuring LDAP

After successful installation, you need to make a password for the admin user using the `ldappassword` command:

```
$ ldappassword
```

The configuration files for OpenLDAP are in `/etc/openldap/slapd.d` directory

You can modify these files directly or use the `ldapmodify` command. It is strongly recommended to modify OpenLDAP using the `ldapmodify` command.

## LDAP Terminology

If we are going to deal with LDAP protocol, there are some terms that we need to know because we will use them a lot.

**Entry** (or object): every unit in LDAP considered an entry.

**dn**: the entry name.

**o**: Organization Name.

**dc**: Domain Component. For example, likegeeks.com is written like this

`dc=likegeeks,dc=com`.

**cn**: Common Name like the person name or name of some object.

## Modifying Entries

OpenLDAP stores its information in bdb or hdb files.

The information saved in the hdb backend can be found in `/etc/openldap/slapd.d/cn=config/olcDatabase={2}hdb.ldif` file.

To identify an element, use the `dn` (distinguished name) attribute. So the first line in our LDIF file will be:

```
dn: olcDatabase={2}hdb,cn=config
```

Then we specify if we want to add or modify

```
changeType: modify
```

We also must clarify if we'll replace it or delete it

replace: olcSuffix

And, finally, we type the new value of the changed attribute.

olcSuffix: dc=likegeeks,dc=local

Back to our file. Change the following entries like this:

olcSuffix: dc=my-domain,dc=com

olcRootDN: cn=Manager,dc=my-domain,dc=com

So our LDIF file will be like this:



```
1 dn: olcDatabase={2}hdb,cn=config
2
3 changeType: modify
4
5 replace: olcSuffix
6
7 olcSuffix: dc=likegeeks,dc=local
8
9 -
10
11 replace: olcRootDN
12
13 olcRootDN: cn=admin,dc=example,dc=local
```

In our file `/etc/openldap/slapd.d/cn=config/olcDatabase={2}hdb.ldif` file. The dn attribute is dn: olcDatabase={2}hdb, and because the file is inside the config folder, so the full dn attribute is dn:

olcDatabase={2}hdb,cn=config

Then we save our file and use `ldapmodify`:

```
$ ldapmodify -Y EXTERNAL -H ldapi:/// -f myfile.ldif
```

```
likegeeks@likegeeks:~  
File Edit View Search Terminal Help  
[root@likegeeks ~]# ldapmodify -Y EXTERNAL -H ldapi:/// -f myfile.ldif  
SASL/EXTERNAL authentication started  
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth  
SASL SSF: 0  
modifying entry "olcDatabase={2}hdb,cn=config"  
[root@likegeeks ~]#
```

You can use the `ldapsearch` command to check the changes:

```
$ ldapsearch -Y EXTERNAL -H ldapi:/// -b cn=config olcDatabase=\*
```

```
likegeeks@likegeeks:/home/likegeeks

File Edit View Search Terminal Help

olcAccess: {0}to * by dn.base="gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth" manage by * none

# {1}monitor, config
dn: olcDatabase={1}monitor,cn=config
objectClass: olcDatabaseConfig
olcDatabase: {1}monitor
olcAccess: {0}to * by dn.base="gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth" read by dn.base="cn=Manager,dc=my-domain,dc=com" read by * none

# {2}hdb, config
dn: olcDatabase={2}hdb,cn=config
objectClass: olcDatabaseConfig
objectClass: olcHdbConfig
olcDatabase: {2}hdb
olcDbDirectory: /var/lib/ldap
olcDbIndex: objectClass eq,pres
olcDbIndex: ou,cn,mail,surname,givenname eq,pres,sub
olcSuffix: dc=example,dc=local
olcRootDN: cn=admin,dc=example,dc=com

# search result
search: 2
result: 0 Success

# numResponses: 5
# numEntries: 4
[root@likegeeks likegeeks]#
```

And yes, the data has been changed.

Also, you can use the `slaptest` command to check the configuration.

```
$ slaptest -u
```

## Adding Entries

To add entries, use the `ldapadd` command.

First, we create our `ldif` file:





```
1 dn: dc=likegeeks,dc=local
2
3 objectClass: dcObject
4
5 objectClass: organization
6
7 dc: likegeeks
8
9 o: likegeeks
```

We specify a series of attributes, like domain component ( dc ), distinguished name ( dn ), and organization ( o ).

According to the type of the object we are creating which is dcObject in our case, some attributes are required, others are optional.

You check the schema according to your system.

On CentOS 6, you can go to `/etc/openldap/slapd.d/cn=config/cn=schema`

On CentOS 7, you can go to `/etc/openldap/schema`

Or you can use grep command to get the .schema files from your system.

The object organization in our example is in `cn={1}core.ldif` file on CentOS 6 or `core.schema` file on CentOS 7.

```
likegeeks@likegeeks:/etc/openldap/schema

File Edit View Search Terminal Help

objectclass ( 2.5.6.4 NAME 'organization'
    DESC 'RFC2256: an organization'
    SUP top STRUCTURAL
    MUST o
    MAY ( userPassword $ searchGuide $ seeAlso $ businessCategory $
        x121Address $ registeredAddress $ destinationIndicator $
        preferredDeliveryMethod $ telexNumber $ teletexTerminalIdentifi
        telephoneNumber $ internationaliSDNNumber $
        facsimileTelephoneNumber $ street $ postOfficeBox $ postalCod
        postalAddress $ physicalDeliveryOfficeName $ st $ l $ descrip

objectclass ( 2.5.6.5 NAME 'organizationalUnit'
    DESC 'RFC2256: an organizational unit'
    SUP top STRUCTURAL
    MUST ou
    MAY ( userPassword $ searchGuide $ seeAlso $ businessCategory $
        x121Address $ registeredAddress $ destinationIndicator $
        preferredDeliveryMethod $ telexNumber $ teletexTerminalIdentifi
        telephoneNumber $ internationaliSDNNumber $
        facsimileTelephoneNumber $ street $ postOfficeBox $ postalCod
        postalAddress $ physicalDeliveryOfficeName $ st $ l $ descrip
```

As we can see, the only required attribute is o which is the organization.

Now we can use the ldapadd command to add our object:

```
$ ldapadd -f myobj.ldif -D cn=admin,dc=likegeeks,dc=local -w mypass
```

We specify the filename using -f, the admin user using -D and the password using -w.

You can check if the entry was created using the ldapsearch command:

```
$ ldapsearch -x -b dc=likegeeks,dc=local
```

## Adding Organizational Units

You can add organizational unit (ou). First, create a new LDIF file. Let's name it users.ldif, and put this:



```
1 dn: ou=users,dc=likegeeks,dc=local
2
3 objectClass: organizationalUnit
4
5 ou: users
```

Then we use ldapadd to add the unit:

```
$ ldapadd -f users.ldif -D cn=admin,dc=likegeeks,dc=local -w mypass
```

This organizational unit holds all LDAP users.

## Adding Users

We can add users to the newly created organizational unit.

First, we create our ldif file:



```
1 dn: cn=adam,ou=users,dc=likegeeks,dc=local
2
3 cn: adam
4
5 sn: USA
6
7 objectClass: myorg
8
9 userPassword: mypass
10
11 uid: adam
```

Then add the user using ldapadd command:

```
$ ldapadd -f adam.ldif -x -D cn=admin,dc=likegeeks,dc=local 2 -w mypass
```

## Adding Groups

Also, we create the ldif file first:



```
1 dn: cn=developers,ou=users,dc=likegeeks,dc=local
2
3 cn: developers
4
5 objectClass: groupOfNames
6
7 member: cn=adam,ou=users,dc=likegeeks,dc=local
```

Then run ldapadd to add the group:

```
$ ldapadd -f groups.ldif -x -D cn=admin,dc=likegeeks,dc=local -w mypass
```

## Deleting Entries

Deleting an entry is very easy, just use ldapdelete command with the cn you want:

```
$ ldapdelete "cn=adam,ou=users,dc=likegeeks,dc=local" -D cn=admin,dc=likegeeks,dc=local -w mypass
```


You can check if the entry is deleted using ldapsearch:

```
$ ldapsearch -x -b "dc=likegeeks,dc=local"
```

## LDAP Port

Ldap port is 389 and in case you secure your ldap using TLS the port will be 636.

You can ensure what port your OpenLDAP is running using the netstat command.

 **ezoic** report this ad\$ netstat -ntlp | grep slapd

```
likegeeks@likegeeks:/home/likegeeks

File Edit View Search Terminal Help

[root@likegeeks likegeeks]# netstat -ntlp | grep slapd
tcp        0          0 0.0.0.0:389          0.0.0.0:*          LISTEN
tcp6       0          0 :::389            :::*               LISTEN

[root@likegeeks likegeeks]#
```

## Authenticating Users with LDAP

By default, Linux authenticates **users** using `/etc/passwd` file. Now we will see how to authenticate users using OpenLDAP.

Make sure you allow the OpenLDAP ports (389, 636) on your system.

If you are using iptables firewall, I recommend you review iptables post to understand these commands **Linux iptables firewall**.

```
$ authconfig --enableldap --enableldapauth --ldapserver ver=192.168.1.10 --
ldapbasedn="dc=likegeeks,dc=local" --enableldaptls --update
```

Now the certificates are in `/etc/openldap/cacerts`.

When we create a user, you have to define some needed fields.

If you want to create a user adam. You will create `adam.ldif` file and write the following:



```
1 dn: uid=adam,ou=users,dc=likegeeks,dc=local
2
3 uid: adam
4
5 cn: adam
6
7 objectClass: account
8
9 objectClass: posixAccount
10
11 objectClass: top
12
13 objectClass: shadowAccount
14
15 userPassword:: mypass
16
17 shadowLastChange: 14846
18
19 shadowMax: 99999
20
21 shadowWarning: 7
22
23 loginShell: /bin/bash
24
25 uidNumber: 500
26
27 gidNumber: 1000
28
29 homeDirectory: /home/adam
```

If you are using CentOS 7 you should encrypt passwords using `slappasswd` command before putting it in your LDIF file like this:

**\$ slappasswd**

Then we copy the encrypted password on the ldif file, so the file will be like this:



```
1 uid: adam
2
3 cn: adam
4
5 objectClass: account
6
7 objectClass: posixAccount
8
9 objectClass: top
10
11 objectClass: shadowAccount
12
13 userPassword:: {SSHA}sathXSo3XTWrbi20Fp+MsqJHjXkHGhT/
14
15 shadowLastChange: 14846
16
17 shadowMax: 99999
18
19 shadowWarning: 7
20
21 loginShell: /bin/bash
22
23 uidNumber: 500
24
25 gidNumber: 1000
26
27 homeDirectory: /home/adam
```

Now we can use `ldapadd` to add the user:

**\$ ldapadd -f adam.ldif -x -D cn=admin,dc=likegeeks,dc=local -w mypass**