UNIX

(Version: 1.4)

Lab Book

Document Revision History

Date	Revision No.	Author	Summary of Changes
	1	Veena Deshpande	New course creation
30-Sept-2009	2	Kishori Khadilkar	Revamped as per new template
20-June-2011	3	Rathnajothi Perumalsamy	Revamped as per Integrated syllabus
30-Sep -2013	4	Amit Sali	Lab for SVN(subversion) is added.

Table of Contents

Document Revision History 2
Table of Contents 3
Getting Started 5
Overview 5
Setup Checklist 5
Instructions 5
Learning More (Bibliography if applicable) 5
Lab 1. Connecting to the Unix Server 6
1.1: Connecting to the Unix Server 6
1.2: Logging out of the system 6
Lab 2. Unix Basic Command 7
2:1 Single Row Functions: 7
Lab 3. UNIX File System & Permissions 10
3.1: Viewing the File System and Granting/Removing Permissions 10
(Note: Create required files if doesn't exists.) 10
Lab 4. JOINS AND SUBQUERIES 11
4.1: Using Pipes and Filters: 11
Lab 5. Vi Editor 14
5.1: Working wth Vi Editor 14
Lab 6. SED Commands 15
6.1: Using SED Commands 15
Lab 7. Process Related Commands 16
7.1: Using Process-Related Commands 16
Lab 8. Shell Script 17
7.1: Writing Shell-Scripts 17
Lab 9. Writing and Executing C and C++ Programs 20
9.1: Write, Compile, Link and Execute a Simple C and C++ 20
Program: 20
Stretched assignments 21
Write AWK scripts for the following: 21
(Use emp.lst file created in Lab 4) 21
1: Find the number of employees belonging to a particular department specified by user 21
2: Find the count of people in each dept. of the employee file 21

3: Generate a list of all S.E. who earn more than the amount specified by user 21
4: View the employee records in order of designations 21
5: List employee details of all employees who earn more than the average salary of all employees. 21
Lab 10. Understanding Makefile and Make Utility 22
10.1: Create and Execute a Makefile: 22
10.2: <TODO> Compiling and executing c++ programs 22
Lab 11. Version Management using subversion (SVN) 24
11.1: Repository Creation 24
11.2: Initial Project Setup 24
11.3: Creating the working copy of project (Check out) 25
11.4: Modifying, updating, creating new file in the project 26
11.5: The lock-modify-unlock Cycle 26
11.6: <TODO> Working on SVN 27

Getting Started

Overview

This lab book is a guided tour for learning Unix. It comprises 'To Do' assignments. Follow the steps provided and work out the 'To Do' assignments.

Setup Checklist

Here is what is expected on your machine in order for the lab to work

Minimum System Requirements

- Intel Pentium 90 or higher (P166 recommended)
- Microsoft Windows 95, 98, or NT 4.0, 2k, XP.
- Memory: 32MB of RAM (64MB or more recommended)

Please ensure that the following is done:

- A text editor like Notepad is installed.
- Participants should be able to connect to UNIX server through telnet (IP address : 192.168.224.34)

Instructions

- For all coding standards refer Appendix A. All lab assignments should refer coding standards.
- Create a directory by your name in drive <drive>. In this directory, create a subdirectory html_assgn. For each lab exercise create a directory as lab <lab number>

Learning More (Bibliography if applicable)

- UNIX Concepts and Application by Sumitabha Das
- The Unix Programming Environment", by Kernighan and Pike.
- UNIX Primer Plus, Third Edition. Don Martin, Stephen Prata, Mitchell Waite, Michael Wessler, and Dan Wilson
- Advanced Unix : a programmer's guide / Stephen Prata

Lab 1. Connecting to the Unix Server

Goals	• Learn to connect to the Unix server • Learn to log out of the Unix server
Time	5 min

1.1: Connecting to the Unix Server

Step 1: Enter your login name and password to login to the UNIX system.

1.2: Logging out of the system

Step 1: Type the exit command at $ prompt or else, press ctrl and d together to log out.

Lab 2. Unix Basic Command

Goals	• Learn to use basic Unix commands
Time	120 min
Lab Setup	Telnet with Unix Server

2:1 Single Row Functions:

1. To display the current working directory, the command is:
   pwd
   The output is as follows.
   /home/trg1
2. Display the path to and name of your HOME directory.
3. Display the login name using which you have logged into the system
4. Display the hidden files of your current directory.
5. List the names of all the files in your home directory.
6. Using the long listing format to display the files in your directory.
7. List the files beginning with chap followed by any number or any lower case alphabet. (Example, it should display all files whose names are like chap1, chap2, chap3 ……., chapa,ahapb,chapc,……..)
8. Give appropriate command to create a directory called C_prog under your home directory. (Note: Check the directory using ls)
9. Create the following directories under your home directory. (Note: Check using ls)

   newdir

   newdirectory
10. List the names of all the files, including the contents of the sub directories under your home directory.
11. Remove the directory called newdirectory from your working directory.
12. Create a directory called temp under your home directory.
13. Remove the directory called newdir under your home directory and verify the above with the help of the directory listing command.
14. Create another directory directorynew under the temp directory.
15. Change the directory to your home directory.
16. From your home directory, change the directory to directorynew using relative and absolute path.
17. Remove the directory called c_prog, which is in your home directory.
18. Change to the directory /etc and display the files present in it.
19. List the names of all the files that begin with a dot in the /usr/bin directory.

20. Create a file first.unix with the following contents.

    Hi! Good Morning everybody.

    Welcome to the First exercise on UNIX.

    Hope you enjoy doing the assignments.

21. Copy the file first.unix in your home directory to first.unics.

    (Note: checked using ls, first.unix file also should exist along with first.unics)

22. List the contents of first.unix and first.unics with a single command.
23. Create a new directory under the temp directory.
24. From your home directory, copy all the files to the directory created under the temp sub directory.
25. Move the file first.unix to the directory temp as second.unix
26. Remove the file called first.unics from the home directory.
27. Change your directory to temp and issue the command rm *. What do you observe?
28. Move all files whose names end with a, c and o to the HOME directory.
29. Copy all files that end with a 'UNIX' to the temp directory.
30. Issuing a single command, remove all the files from the directory temp and the directory itself.
31. Try commands cp and mv with invalid number of arguments and note the results.
32. Use the cat command to create a file friends, with the following data:

Madhu	6966456	09/07/68
Jamil	2345215	08/09/67
Ajay	5546785	01/04/66
Mano	7820022	09/07/68
David	8281292	09/09/60
Simmi	7864563	12/12/70
Navin	2224311	30/05/68

    The fields should be separated by a tab.

33. Display contents of the file friends.
34. Copy contents of friends to newfriend without using the cp command.
35. Display contents of the file friends and newfriends in a single command.
36. Find all users currently working on the system and store the output in a file named as users.
37. Append contents of friends file to the file, users.
38. Display current system date and time and record your observations. How is the time displayed?
39. Display calendar for the month and year of your birth.
40. Try following commands and record your observations.

    date "+ %"

date “+%m”

date “+%D”

date “+%/%Training Activity”

date “+%Training Activity”

date “+%r”

Lab 3. UNIX File System & Permissions

Goals	• Learn to grant and to remove permissions and to view the file system
Time **Lab Setup**	15 min Telnet with Unix Server

3.1: Viewing the File System and Granting/Removing Permissions

(Note: Create required files if doesn't exists.)

1. Give the execute permission for the user for a file chap1
2. Give the execute permission for user, group and others for a file add.c
3. Remove the execute permission from user, give read permission to group and others for a file aa.c
4. Give execute permission for users for a.c, kk.c, nato and myfile using single command
5. Change the directory to root directory. Check the system directories, like bin, etc, usr etc

Lab 4. JOINS AND SUBQUERIES

Goals	• Learn to use Pipes & Filters in UNIX
Time	120 min
Lab Setup	Telnet with Unix Server

4.1: Using Pipes and Filters:

1: Redirect the content of the help document ls, into a file called as lsdoc.

2: Display the content of the lsdoc page wise.

3: Display only the first 4 lines of the lsdoc file.

4: Display only the last 7 lines of the file lsdoc.

5: Remove the file lsdoc.

6: There will be B'day celebration from the friends file, find how many B'day parties will be held. If two of the friends have the B'date on the same day, then we will be having one party on that day.

7: Display the lines starting with Ma, in the file friends.

8: Display the lines starting with Ma, ending with i or ending with id, in the file friends.

9: Print all the files and the directory files from the current directory across all the sub directories, along with its path

10: Print only the Directory files.

11: Display the files starting with chap, along with its path.

12: Sort the file friends in ascending order of names.

13: Display the contents of the file friends in uppercase letters.

14: Store the contents of your home directory in a file called dir.

15: From the above file dir, display the file permissions and the name of the file only.

16: From the same dir file, store only the file names in a file called files.

17: From the same dir file, store only the permissions of files in a file called perms.

18: From the same dir file, store only the file sizes in a file called sizes.

19: Display the file names, sizes and permissions from your directory in that order.

20: Display the number of users working on the system.

21: Find out the smallest file in your directory.

22: Display the total number of lines present in the file friends.

23:Create the following fixed record format files (with "|" delimiter between fields) with the structure given below, and populate them with relevant data use these files to solve following questions

emp.lst: Empid(4),Name(18),Designation(9),Dept(10),Date of Birth(8),Salary(5)

dept.lst : Dept.Code(2),Name(10),Head of Dept's id(4)

desig.lst: Designation Abbr.(2), Name (9)

1. Find the record lengths of each file.
2. Display only the date of birth and salary of the last employee record.
3. Extract only employee names and designations. (Use column specifications). Save output as cfile1.
4. Extract Emp.id, dept, dob and salary. (Use field specifications). Save output as cfile2.
5. Fix the files cfile1 and cfile2 laterally, along with the delimiter.
6. Sort the emp.lst file in reverse order of Emp. Names.
7. Sort the emp.lst file on the salary field, and store the result in file srtf.
8. Sort the emp.lst file on designation followed by name.
9. Sort the emp.lst file on the year of birth.
10. Find out the various designations in the employee file. Eliminate duplicate listing of designations.
11. Find the non-repeated designation in the employee file.
12. Find the number of employees with various designations in the employee file.
13. Create a listing of the years in which employees were born in, along with number of employees born in that year.
14. Use nl command to create a code table for designations to include designation code (Start with dept. code 100, and subsequently 105, 110 …).

24: PCS has its offices at Pune, TTC and Mumbai. The employees' data is stored separately for each office. Create appropriate files (with same record structure as in previous assignment) and populate with relevant data.

1. List details about an employee 'Manu Sharma' in the Mumbai office.

2. List only the Emp.Id. And Dept. of Manu Sharma.
3. List details of all managers in all offices. (O/P should not contain file names.).
4. Find the number of S.E. in each office.
5. List only the Line Numbers and Employee names of employees in 'H/W' in Pune file.
6. Obtain a listing of all employees other than those in 'HR' in the Mumbai file and save contents in a file 'nonhr'.
7. Find the name and designation of the youngest person who is not a manager.
8. Display only the filename(s) in which details of employee by the name 'Seema Sharma' can be found.
9. Locate the lines containing saxena and saksena in the Mumbai office.
10. Find the number of managers who earn between 50000 and 99999 in the Pune office.
11. List names of employees whose id is in the range 2000 – 2999: in Pune Office; in all offices.
12. Locate people having same month of birth as current month in Pune office.
13. List details of all employees other than those of HR and Admin in file F1.
14. Locate for all Dwivedi, Trivedi, Chaturvedi in Pune file.
15. Obtain a list of people in HR, Admin and Recr. depts. sorted in reverse order of the dept.

Stretched assignments:

25: Write a command sequence that prints out date information in this order: time, day of week, day number, month, year:
13:44:42 IST Sun 16 Sept 1994

26: Write a command sequence that prints the names of the files in the current directory in the descending order of number of links

27: Write a command sequence that prints only names of files in current working directory in alphabetical order

28: Write a command sequence to print names and sizes of all the files in current working directory in order of size

29: Determine the latest file updated by the user

Lab 5. Vi Editor

Goals	Work with Vi Editor in Unix
Time	30 min
Lab Setup	Telnet with Unix Server

5.1: Working wth Vi Editor

1. Create a file using Vi. Enter the following text:
   A network is a group of computers that can communicate with each other, share resources, and access remote hosts or other networks. Netware is a computer network operating system designed to connect, manage, and maintain a network and its services. Some of the network services are Netware Directory Services (NDS), file system, printing and security.

   a. Change the word “Netware” in the second line to “Novell Netware”.
   b. Insert the text “(such as hard disks and printers)” after “share resources” in the first line.
   c. Append the following text to the file:
   “Managing NDS is a fundamental administrator role because NDS provides a single point for accessing and managing most network resources.”

2: Create the data files, used in the previous lab sessions using vi editor.

Lab 6. SED Commands

Goals	Learn to use SED Commands in Unix
Time	15 min
Lab Setup	Telnet with Unix Server

6.1: Using SED Commands

1. Create a file “Employee.dat” with text as follows.

James	76382	PACE	Chennai
John	34228	GRIT	Hyderabad
Peter	22321	GE	Bangalore
Albert	32342	GRIT	Pune
Mathew	23222	PACE	Mumbai
Richard	23232	ACS	Pune

a) Write a sed command to print only the lines starting at line 2 and ending with the letters “Pune”

b) Write a sed command that will display the top 5 lines from the file

c) Write a sed command that will substitute the word “Chennai” for "Pune" used in all instance of the word

d) Write a sed command that will replace occurrence of the character e with the string UNIX in all lines. (Use –e option)

e) Write a sed command to delete blank lines

f) Write a sed command to delete lines from 3 to 5

2**: Create a new file “PACE.dat which has only the lines that contain the word “PACE” from Employee.dat**

Lab 7. Process Related Commands

Goals	Learn to use process-related commands in Unix
Time	15 min
Lab Setup	Telnet with Unix Server

7.1: Using Process-Related Commands

1. Find out the PID of the processes that are activated by you
2. Find out the information about all the processes that are currently active
3. Start a different process in the background. Find out the status of the background process using the PID of the same.

Lab 8. Shell Script

Goals	Learn to write simple shell scripts
Time	3 Hrs
Lab Setup	Telnet with Unix Server

7.1: Writing Shell-Scripts

1. Display the Primary and Secondary prompt. Change the primary prompt to your name: temporarily

2: As soon as you login, the prompt should be changed to your name: also the name of the home directory should be automatically displayed.

3: Check the content of the Environmental variable SHELL.

4: Try the below exercise and check the output.

Note: Type every line and press enter, do not type the entire code in a vi editor.

```
$continent=”Africa”
$echo “$continent”
                ----------→ Africa
$sh
$echo “$continent”
                ----------→ No Response
$continent=”Asia”
$echo “$continent”
                ----------→ Asia
$ctrl + d
$echo “$continent”
                ----------→ Africa
$sh
$echo “$continent”
                ----------→ No Response
$ctrl + d
```

5: Try the below exercise and check the output. (Export variables)

Note: Type every line and press enter, do not type the entire code in a vi editor.

```
$continent=”Africa”
export continent
$echo “$continent”
                ----------→ Africa
```

```
$sh
$echo “$continent”
                    ----------→ Africa
$continent=”Asia”
$echo “$continent”
                    ----------→ Asia
$ctrl + d
$echo “$continent”
                    ----------→ Africa
```

6: Write a shell script that takes the user name as input and reports whether he / she has logged in or not.

7: Write a shell script to display the file name and its contents of all the files that is there in the current directory.

8: Write a shell script, which will take a file name as argument and check whether the file exists and display its access permissions for user.

9: Pass three numbers as command line arguments and display the largest number in the given three numbers.

10: Write a shell script which will accept a pattern and a file name. The pattern will be searched in the file provided. Display appropriate messages and perform necessary validations on file.

11: To create a menu program for a) creating a file, b) Creating a directory, c) copying a file, d) moving a file. (use functions)

- **a.** If the file exists already give the appropriate message
- **b.** If the dir exists already give the appropriate error message
- **c.** Source file should exist if not give a message, It should have read permission if not another message, Destination file either there or not, if not there then create it and copy it. If there, then ask whether to overwrite or not, if yes then overwrite it or else give a message file exists already and not overwritten.

12: Write a function yesno() to display question to user and accept answer as y/n. If answer to the question is y the function should return 0 otherwise 1.

Use yesno functions for asking different questions. Question will be passed as parameter to the function.

Accept filename from user check whether it is file or directory. Use yesno() function to display question do you really want to delete file? If the ans is y, then delete the file or directory.

13: Write a shell script to store names of four employees and check whether those employees are currently logged in or not. Display appropriate message.

14: Accept the user's first and last name and the echo the entire name along with some suitable comment.

15: List all files that have been modified today.

16: Display long listing of only the regular files in the current directory.

17: Display details of all files in the 2 “paths” accepted from user. The display should be screen by screen.

18: Let the script display its name and its PID.

19: Get the concatenated o/p of 2 files into a third file: Take 3 command line arguments: The first argument is the name of a destination file, and the other two arguments are names of files whose contents are to be placed in the destination file.

Stretched Assignments:

20: Write a menu driven shell program to:
a. Display calendar of current month
b. Search for a pattern in all the files/subdirectories from current directory.
c. Count the no. of directories / sub directories in current directory

21: Display day of week for a given date. (ddmmyyyy)
If day is Monday, display message “Monday Blues”
Friday display message “yeh! It’s week end.”
Similarly display different messages for each day of the week.

22: Display the contents of all .lst files in the current directory.

23: Design a simple calculator, which will add/subtract/multiply/divide 2 numbers.
eg. cal 10 20 + will give o/p as 30.

24: For a student file with the following fields, rollno, name, marks, Generate 2 files ‘Pass’ and ‘Fail’ containing records of student who have passed or failed. Also count the number of students who have passed or failed.

25: Accept a date string from terminal and display employees born after the input date.

Lab 9. Writing and Executing C and C++ Programs

Goals	• Learn to use process-related commands in Unix Learn how to write, compile, link and execute simple C and C++ programs • Understand the steps involved from writing program to executing it
Time	60 min
Lab Setup	Linux Operating System and gcc and g++ compilers

9.1: Write, Compile, Link and Execute a Simple C and C++

Program:

Step 1: Login to the **Linux Operating** System.

Login to the home directory. Use the login details provided by faculty to login in the system

Step 2: Use vi editor to write following C programs:

pi.c

```
#include <math.h>

mypi()
{
printf("pi = %.5f\n", 4 * atan(1.0));
}
```

hello.c

```
#include <stdio.h>
main ()
{
mypi();
}
```

Step 3: Compile the source files.

Use the gcc command to compile pi.c and hello.c source files to create pi.o and hello.o object files.

```
$ gcc -c pi.c
$ gcc -c hello.c
```

Step 4: Link the object files.

Execute the gcc command with –lm –o command option to link the object files to create executable file hello (-lm option will include library maths).

```
$ gcc -lm -o hello hello.o pi.o
```

Step 5: Execute the hello file.

Execute the hello program to see the output by giving following command (./ indicate search hello file in current directory otherwise the file will be searched in bin directory)

```
$  ./hello
pi = 3.14159
```

Stretched assignments

Write AWK scripts for the following:

(Use emp.lst file created in Lab 4)

1: Find the number of employees belonging to a particular department specified by user

2: Find the count of people in each dept. of the employee file

3: Generate a list of all S.E. who earn more than the amount specified by user

4: View the employee records in order of designations

5: List employee details of all employees who earn more than the average salary of all employees.

Lab 10. Understanding Makefile and Make Utility

Goals	• Learn how to create a makefile to automate the build process • Understand the steps involved in creating and executing Makefile using make utility
Time	60 min
Lab Setup	Linux Operating System and gcc and g++ compilers

10.1: Create and Execute a Makefile:

1. Create a make file.
2. Using vi editor create Makefile as shown below:

```
$ cat Makefile
pi.o:   pi.c
        gcc -c pi.c

hello.o: hello.c
        gcc -c hello.c

hello:  hello.o pi.o
         gcc -lm -o hello hello.o pi.o
```

Note: You need TABS after each ":" and before the gcc on the lines that follow each "rule". Basically, a makefile says "to make hello file, you need hello.o and pi.o files, and you run this command"

Step 2: Execute Makefile.

3. To execute Makefile use the make command as shown below (in the command hello indicates the target):

```
$ make hello
gcc -c hello.c
gcc -c pi.c
gcc -lm -o hello hello.o pi.o
$ ./hello
pi = 3.14159
$
```

Note: Leave out tabs in Makefile, make will complain:

Makefile:5: * missing separator. Stop**

10.2: <TODO> Compiling and executing c++ programs

Write picpp.cpp and hellocpp.cpp c++ programs to do same task as that of pi.c and hello.c respectively.

Write Makefile1 to build hellocpp executable file and execute hellocpp executable program.

Note: You need to use g++ compiler to compile the C++ program. Also explore how to invoke Makefile1 from make utility.

Lab 11. Version Management using subversion (SVN)

Goals	• Learn the concept of version management • Understand the steps involved in version management using SVN as a tool
Time	60 min
Lab Setup	Linux Operating System and SVN client

11.1: Repository Creation

This is a Linux Administrator task and assumes that he has created the repository at location "**/home/demo_project_repository**".

11.2: Initial Project Setup

Option 1: Using System administrator/ Normal User (If the project is newly started)

Creating a folder

```
[root@pace ~]# svn mkdir
file:///home/demo_project_repository/Analysis -m "Creating
Project Analysis directory"

Committed revision 1.
```

Deleting a folder

```
[root@pace ~]# svn rm
file:///home/demo_project_repository/Analysis -m "Removing
Project Analysis directory"

Committed revision 2.
```

Checking log file

```
[root@pace ~]# svn log file:///home/demo_project_repository
------------------------------------------------------------------------
r2 | root | 2013-09-30 09:47:17 +0530 (Mon, 30 Sep 2013)| 1 line
Removing Project Analysis directory
------------------------------------------------------------------------
r1 | root | 2013-09-30 09:46:49 +0530 (Mon, 30 Sep 2013)| 1 line
Creating Project Analysis directory
------------------------------------------------------------------------
```

Option 2: Using client (if the directory structure of project is already defined)

Note: - This activity is call project import

Initial project directory structure at client side

```
[testuser1@pace project]$ tree -a
.
|-- Analysis
|   |-- ReadMe.txt
|   `-- Software_Requirement_Specification.docx
|-- Coding
|-- Design
|   |-- High_Level_Design_1.2.docx
|   `-- Low_Level_Design_1.2.docx
|-- Query_Tracking_Sheet
|-- Testing
|   `-- test_case-lv1.0.xls
`-- Timesheet
```

Importing the Initial Project to Repository

```
[testuser1@pace project]$ svn import /home/testuser1/project
file:///home/demo_project_repository -m 'Initial import'
Adding         /home/testuser1/project/Query_Tracking_Sheet
Adding         /home/testuser1/project/Coding
Adding         /home/testuser1/project/Analysis
Adding
/home/testuser1/project/Analysis/Software_Requirement_Specificat
ion.docx
Adding         /home/testuser1/project/Analysis/ReadMe.txt
Adding         /home/testuser1/project/Timesheet
Adding         /home/testuser1/project/Design
Adding
/home/testuser1/project/Design/High_Level_Design_1.2.docx
Adding
/home/testuser1/project/Design/Low_Level_Design_1.2.docx
Adding         /home/testuser1/project/Testing
Adding         /home/testuser1/project/Testing/test_case-
lv1.0.xls

Committed revision 3.
```

The project was imported to the SVN, now delete the local directories

11.3: Creating the working copy of project (Check out)

Login to the client, go to the folder where working copy is to be created and fire the following commands

```
[testuser2@pace project]$ ls -a
.  ..
[testuser2@pace project]$ svn checkout
file:///home/demo_project_repository
A    demo_project_repository/Query_Tracking_Sheet
A    demo_project_repository/Analysis
A
demo_project_repository/Analysis/Software_Requirement_Specificat
ion.docx
A    demo_project_repository/Analysis/ReadMe.txt
```

```
A    demo_project_repository/Coding
A    demo_project_repository/Timesheet
A    demo_project_repository/Design
A    demo_project_repository/Design/High_Level_Design_1.2.docx
A    demo_project_repository/Design/Low_Level_Design_1.2.docx
A    demo_project_repository/Testing
A    demo_project_repository/Testing/test_case-lv1.0.xls
Checked out revision 3.
[testuser2@pace project]$ tree ./demo_project_repository
./demo_project_repository
|-- Analysis
|   |-- ReadMe.txt
|   `-- Software_Requirement_Specification.docx
|-- Coding
|-- Design
|   |-- High_Level_Design_1.2.docx
|   `-- Low_Level_Design_1.2.docx
|-- Query_Tracking_Sheet
|-- Testing
|   `-- test_case-lv1.0.xls
`-- Timesheet
```

11.4: Modifying, updating, creating new file in the project

Login to the client, go to the folder where working copy is stored, do the changes and fire the following commands

```
[testuser2@pace demo_project_repository]$ cd Analysis
[testuser2@pace Analysis]$ touch SignOff.doc
[testuser2@pace Analysis]$ touch Open_Item.doc
[testuser2@pace Analysis]$ vi ReadMe.txt
[testuser2@pace Analysis]$ svn status
?       SignOff.doc
?       Open_Item.doc
M       ReadMe.txt
[testuser2@pace Analysis]$ cd ..
[testuser2@pace demo_project_repository]$ svn add
Analysis/SignOff.doc Analysis/Open_Item.doc
A         Analysis/SignOff.doc
A         Analysis/Open_Item.doc
[testuser2@pace demo_project_repository]$ svn commit -m 'User
Confirmation Document Added'
Adding         Analysis/Open_Item.doc
Sending        Analysis/ReadMe.txt
Adding         Analysis/SignOff.doc
Transmitting file data ...
Committed revision 4.
```

11.5: The lock-modify-unlock Cycle

Login to the client with **user1**, go to the folder where working copy is stored, lock the file on which **user1** want to work

```
[testuser1@pace demo_project_repository]$ svn lock
Analysis/ReadMe.txt -m "Locking a File for Change"
'ReadMe.txt' locked by user 'testuser1'.
```

Login to the client with **user2**, go to the folder where working copy is stored, try to change the same file which has been locked by user1.

```
[testuser2@pace demo_project_repository]$ cd Analysis
[testuser2@pace Analysis]$ vi ReadMe.txt
[testuser2@pace Analysis]$ svn commit -m 'User Confirmation
Document Added'
Sending        Analysis/ReadMe.txt
Transmitting file data .svn: Commit failed (details follow):
svn: User testuser2 does not own lock on path
'/Analysis/ReadMe.txt' (currently locked by testuser1)
```

Now user1 can either perform the changes in the file and commit it (This will automatically release the lock) or just unlock the file so that user2 will start working on it.

```
[testuser1@pace demo_project_repository]$ svn unlock
Analysis/ReadMe.txt
'ReadMe.txt' unlocked.
```

Now user2 can perform the changes and commit it.

```
[testuser2@pace Analysis]$ svn commit -m 'User Confirmation
Document Added'
Sending        Analysis/ReadMe.txt
Transmitting file data .
```

11.6: <TODO> Working on SVN

1) Create four folders in SVN repository with following structure

```
|-- YourUserName
|   |-- Research
|   |-- Development
|   |-- RND
```

2) Remove the directory RND from the repository
3) Add two files code.txt under folder Research and process.doc under folder Development
4) Lock file code.txt , change the contents of the file and update the changes to SVN server