Lab Manual

Unix and Linux Programming (Pr)

COT-218 and IT-214

Unix and Linux Programming
COT-210

L	T	P
3	1	-

1. Linux Startup
User accounts, accessing linux – starting and shutting processes, Logging in and Logging out, Command line, simple commands

2. Shell Programming
Unix file system: Linux/Unix files, inodes and structure and file system related commands, Shell as command processor, shell variables, creating command substitution, scripts, functions, conditionals, loops, customizing environment

3. Regular Expressions and Filters
Introducing regular expressions patterns, syntax, character classes, quantifiers, introduction to egrep, sed, programming with awk and perl.

4. The C Environment
The C compiler, vi editor, compiler options, managing projects, memory management, use of makefiles, dependency calculations, memory management – dynamic and static memory, building and using static and dynamic libraries, using ldd, soname, dynamic loader, debugging with gdb

5. Processes in Linux
Processes, starting and stopping processes, initialization processes, rc and init files, job control – at, batch, cron, time, network files, security, privileges, authentication, password administration, archiving, Signals and signal handlers, Linux I/O system

Books

1. John Goerzen: Linux Programming Bible, IDG Books, New Delhi, 2000.
2. Sumitabha Das: Your Unix – The Ultimate Guide, TMH, 2000.
3. Mathew: Professional Linux Programming, vol.1 & 2, Wrox-Shroff, 2001.
4. Welsh & Kaufmann: Running Linux, O’Reiley & Associates, 2000.

Unix and Linux Programming
IT-204

L	T	End Semester:	60
3	1	Mid Semester:	40

1. Linux Startup

User accounts, accessing Linux – starting and shutting processes, Logging in and Logging out, Command line, simple commands

2. Shell Programming

Unix file system: Linux/Unix files, inodes and structure and file system related commands, Shell as command processor, shell variables, creating command substitution, scripts, functions, conditionals, loops, customizing environment

3. Regular Expressions and Filters

Introducing regular expressions patterns, syntax, character classes, quantifiers, introduction to egrep, sed, programming with awk and perl.

4. The C Environment

The C compiler, vi editor, compiler options, managing projects, memory management, use of makefiles, dependency calculations, memory management – dynamic and static memory, building and using static and dynamic libraries, using ldd, soname, dynamic loader, debugging with gdb

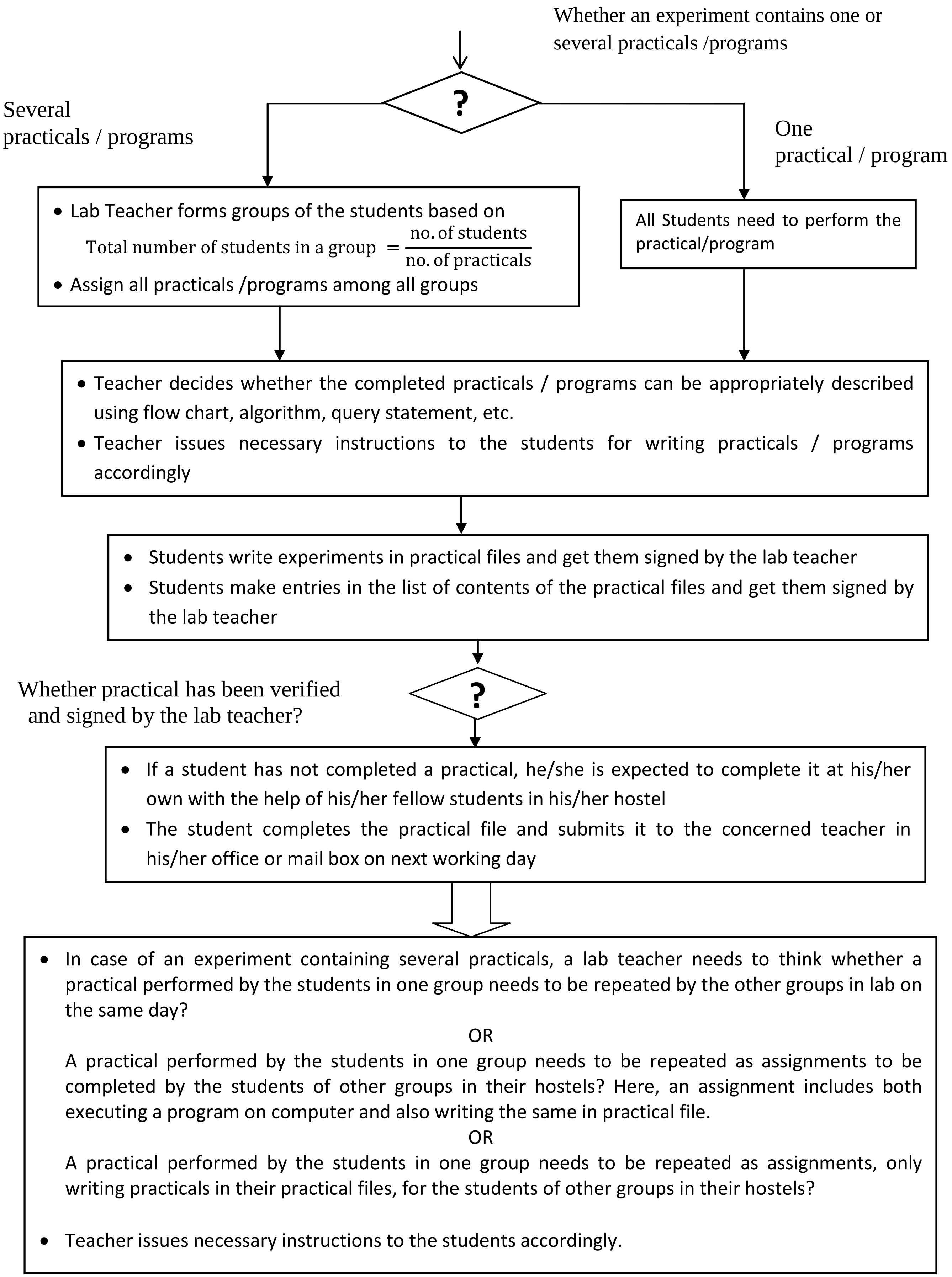
5. Processes in Linux

Processes, starting and stopping processes, initialization processes, rc and init files, job control – at, batch, cron, time, network files, security, privileges, authentication, password administration, archiving, Signals and signal handlers, Linux I/O system

BOOKS

1. John Goerzen: Linux Programming Bible, IDG Books, New Delhi, 2000.
2. Sumitabha Das: Your Unix – The Ultimate Guide, TMH, 2000.
3. Mathew: Professional Linux Programming, vol.1 & 2, Wrox-Shroff, 2001.
4. Welsh & Kaufmann: Running Linux, O'Reiley & Associates, 2000.

Lab Instructions



Whether an experiment contains one or several practicals /programs

?

Several practicals / programs

- Lab Teacher forms groups of the students based on

  $$\text{Total number of students in a group} = \frac{\text{no. of students}}{\text{no. of practicals}}$$

- Assign all practicals /programs among all groups

One practical / program

All Students need to perform the practical/program

- Teacher decides whether the completed practicals / programs can be appropriately described using flow chart, algorithm, query statement, etc.
- Teacher issues necessary instructions to the students for writing practicals / programs accordingly

- Students write experiments in practical files and get them signed by the lab teacher
- Students make entries in the list of contents of the practical files and get them signed by the lab teacher

Whether practical has been verified and signed by the lab teacher?

?

- If a student has not completed a practical, he/she is expected to complete it at his/her own with the help of his/her fellow students in his/her hostel
- The student completes the practical file and submits it to the concerned teacher in his/her office or mail box on next working day

- In case of an experiment containing several practicals, a lab teacher needs to think whether a practical performed by the students in one group needs to be repeated by the other groups in lab on the same day?

  OR

  A practical performed by the students in one group needs to be repeated as assignments to be completed by the students of other groups in their hostels? Here, an assignment includes both executing a program on computer and also writing the same in practical file.

  OR

  A practical performed by the students in one group needs to be repeated as assignments, only writing practicals in their practical files, for the students of other groups in their hostels?

- Teacher issues necessary instructions to the students accordingly.

Lab Manual

Unix and Linux Programming (Pr)

COT-218 and IT-214

L	T	P	
-	-	3	

Practical exam: 40
Sessional: 60

Prerequisite for further experiments: Two hour lecture on LINUX

Refer: http://spoken-tutorial.org/tutorial-search/?search_foss=Linux&search_language=English
http://www.redhat.com/en/files/resources/en-rhel-whats-new-in-rhel-712030417.pdf

Experiment 1 (Basic commands)

I. a) Installation of Unix/Linux operating system.

b) Study of logging/logout details.

c) Study of Unix/Linux general purpose utility command list obtained from (man, who, cat, cd, cp, ps, ls, mv, rm, mkdir, rmdir, echo, more, date, time, kill, history, chmod, chown, finger, pwd, cal, logout, shutdown) commands.

d) Study of vi editor.(http://www.tutorialspoint.com/unix/pdf/unix-vi-editor.pdf)

e) Study of Bash shell, Bourne shell and C shell in Unix/Linux operating system.

f) Study of Unix/Linux file system (tree structure).

g) Study of .bashrc, /etc/bashrc and Environment variables.

Experiment 2 (C programs)

I. Write a C program to check whether the given string is palindrome or not using Command line substitution.

II. Write a C program to emulate the UNIX ls-l command.

III. Write a C program to check the given integer is prime or not.

IV. Write a C program to display Largest of three numbers.

V. Write a C program to check whether the given number is Avogadro number or not.

VI. Write a C program to find the Factorial of a given number.

VII. Write a C program that accept two integers as its arguments and computes the value of first number raised to the power of second number.

Experiment 3 (Shell scripts)

I. Write a shell script program to display list of user currently logged in.

II. Write a shell script program to display “HELLO WORLD”.

III. Write a shell script program to develop a scientific calculator.

IV. Write a shell Script program to check whether the given number is even or odd.

V. Shell script Program to search whether element is present is in the list or not.

Experiment 4 (Shell scripts and sed)

I. Shell script program to check whether given file is a directory or not.

II. Shell script program to count number of files in a Directory.

III. Shell script program to copy contents of one file to another.

IV. Create directory, write contents on that and Copy to a suitable location in your home directory.

V. Use a pipeline and command substitution to set the length of a line in file to a variable.

VI. Write a program using sed command to print duplicated lines of Input.

Experiment 5 (grep, awk, perl scripts)

I. (a) Write a grep/egrep script to find the number of words character, words and lines in a file.

(b) Write an awk script to develop a Fibonacci series.

II. (a) Write a perl script to compute the n^{th} power of a given number.

(b) Write an awk script to display the pattern of given string or number.

III. (a) Write a perl script to check a number is prime or not.

(b) Write an egrep script to display list of files in the directory.

Experiment 6 (programming)

I. Write a shell script program to display the process attributes.

II. Write a shell script to change the priority of processes.

III. Write a shell script to change the ownership of processes.

IV. Write a program to send back a process from foreground.

V. Write a program to retrieve a process from background.

VI. Write a program to create a Zombie process.

VII. Write a program to create a child process and allow the parent to display “parent” and the child to display “child” on the screen.

Experiment 7 (Shell script programming)

I. Write a shell script program to check variable attributes of file and processes.

II. Write a shell script program to check and list attributes of processes.

III. Shell Script program to implement read, write, and execute permissions.

IV. Shell Script program for changing process priority.

Experiment 8 (gdb)

I. To execute programs using gdb to utilize its various features like breakpoints, conditional breakpoints.

II. Write a shell script program to include verbose Debug option for debugging.

III. Write a shell script program to include xtrace Debug option for debugging.

IV. Write a shell script program to include verbose and trace Debug option for debugging.

Experiment 9 (Installations)

I. Installation of VirtualBox (VMWare) on a PC having other operating system.

II. Installation of Cygwin on a PC having other operating system.

III. Installation of NS2 on a PC having Unix/Linux operating system.

IV. Installation of Unix/Linux packages.