Laboratory Manual

for

Advanced Operating System Concepts
(MF 203)

M.Tech (IT) SEM - II

June 2011

Faculty of Technology
Department of Information Technology
Dharmsinh Desai University Nadiad.
www.ddu.ac.in

Dharmsinh Desai University, Nadiad
Faculty of Technology
Department of Information Technology
Laboratory Manual
M.Tech. – IT, Sem: 2, Subject Name: Advanced Operating System Concepts

List of Experiments:

EXPERIMENT 1: Study of various system calls of Linux/Unix System

EXPERIMENT 2: Write a program called executer that takes another command as an argument and executes that command. At the end of execution of the specified command, the executer should display relevant statistics.

EXPERIMENT 3: Using the make utility and the gdb tool for program development

EXPERIMENT 4:
Process tracing
(i) Trace process using strace
(ii) Trace process using ltrace

EXPERIMENT 5: Study Multiprocessor Process Scheduling using LEKIN software system

EXPERIMENT 6: Multi-core programming using OpenMP

PROJECT (The work is equivalent to 4 Lab Experiments)
Each student has to work either individual or in a group of two students on the project assigned.

LABWORK BEYOND CURRICULA

EXPERIMENT 1: Print i-node information using stat/fstat. Also identify type of a file (device file, pipe, directory, link etc.)

EXPERIMENT 2: Implement copy command that can copy source directory to destination, including all files and subdirectories of the source directory.

Dharmsinh Desai University, Nadiad
Faculty of Technology
Department of Information Technology
Laboratory Manual
M.Tech. – IT, Sem: 2, Subject Name: Advanced Operating System Concepts

COMMON PROCEDURE
The common procedure for solving programming related problems is as follows:

Step 1: For given problem statement, find out the names of required system calls.
Step 2: Study and understand the usage of those system calls.
Step 3: Design the logic for solving the given problem.
Step 4: Implement the logic in C programming code.
Step 5: Compile the program using make utility
Step 6: Run the program by passing the appropriate command line arguments
Step 7: Note down the output and draw your conclusion.

EXPERIMENT 1:

Aim: Study of various system calls of Linux/Unix System

Tools / Apparatus: Linux OS and man pages

Procedure:
1. For each command, read the documentation from man pages, e.g., using man command.
2. Study important options
3. Execute the command with options and study the output.

EXPERIMENT 2:

Aim: Write a program called executer that takes another command as an argument and executes that command. At the end of execution
of the specified command, the executer should display following statistics.
- The amount of CPU time used (both user and system time) (in milliseconds),
- The elapsed "wall-clock" time for the command to execute (in milliseconds),
- The number of times the process was preempted involuntarily (e.g. time slice expired, preemption by higher priority process),
- The number of times the process gave up the CPU voluntarily (e.g. waiting for a resource),
- The number of page faults

- The number of page faults that could be satisfied from the kernel's internal cache (e.g. did not require any input/output operations).

  For example:
  Running % executer cat /home/user/test.c

invokes the cat command on the file /home/user/test.c, which will print the content of test.c file. And then displays statistics showing utilization of some system resources.

Hint:
Following system calls will be useful
fork()
getrusage()
gettimeofday()
execve()
wait()
chdir()
strtok()

Tools / Apparatus: Linux OS and gcc

EXPERIMENT 3:

Aim: Using the make utility and the gdb tool for program development

Tools / Apparatus: Linux OS, make, gcc, gdb

Procedure:
Use common procedure

EXPERIMENT 4:

Aim: Process tracing
(i) Trace process using strace
(ii) Trace process using ltrace

Tools / Apparatus: Linux OS, strace, ltrace

Procedure:
1. Run any linux command using strace and ltrace
2. Note down system calls/functions used by the command
3. Write down your own program and compile it using gcc
4. Run your executable program using strace and ltrace

5. Write down system calls/functions used by your executable program.

EXPERIMENT 5:

Aim: Study Multiprocessor Process Scheduling using LEKIN software system

Tools / Apparatus: Windows OS, LEKIN software

Procedure:
1. Using GUI facility of LEKIN provide configuration of a hypothetical multiprocessor system.
2. Using GUI facility of LEKIN, provide configuration of processes with CPU burst.
3. Simulate execution of the processes on the hypothetical multiprocessor system.
4. Compare Gantt chart of process execution for various scheduling algorithms.

EXPERIMENT 6:

Aim: Multi-core programming using OpenMP

Tools / Apparatus: Linux OS, make, gcc with support of OpenMP

Procedure:
Use common procedure

PROJECT (The work is equivalent to 4 Lab Experiments)
Each student has to work either individual or in a group of two students on the project assigned.

List of Projects
1. Process Exlporer using proc (Implementation: C language)
2. System Explorer using proc (Implementation: C language)
3. Implementation of pstree command (Implementation: C language)
4. Implementation of a new system call (Implementation: C language)
5. System call tracer-strace (Implementation: C language)
6. Detectction of USB devices (Implementation: C language)
7. Implementation of Emulated Linux File system on windows (C or Java)
8. Process Exlporer using proc (Implementation: Java language)

9. System Explorer using proc (Implementation: Java language)

10. Implementation of pstree command (Implementation: Java language)

11. Modifying scheduling code of kernel

12. Implementation of Device Driver

13. Implementation of Command Shell

Study Projects with some implementation or demonstration (installation and testing)

1. Study of Linux ELF Loader
2. Study of integrating Bare Metal Hypervisor, Satorage Area Network (SAN), and Network Attached Storage (SAN)

LABWORK BEYOND CURRICULA

EXPERIMENT 1:

Aim: Print i-node information using stat/fstat. Also identify type of a file (device file, pipe, directory, link etc.)

Tools / Apparatus: Linux OS, gcc

Procedure:
Use common procedure

EXPERIMENT 2:

Aim: Implement copy command that can copy source directory to destination, including all files and subdirectories of the source directory.

Tools / Apparatus: Linux OS

Procedure:
Use common procedure

Required Reading Materials
Books:

Unix Programming environment By: Kernighan and Pike.