virtusa®
Accelerating Business Outcomes

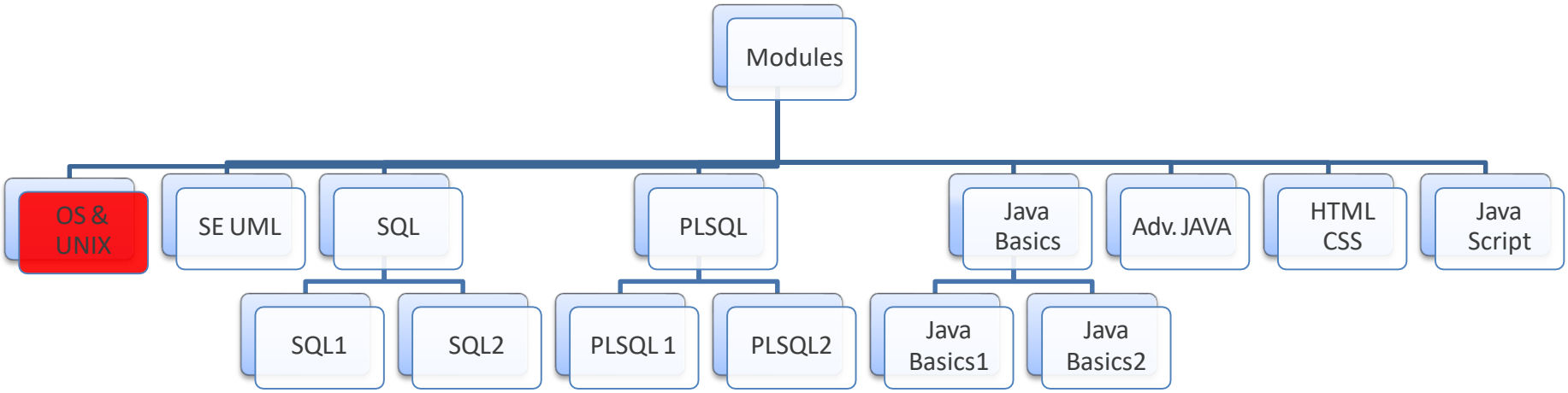
Operating System and UNIX

Module Overview

Purpose:

The following modules are identified to build the basic IT skills and acquaint you with the technology basics.

The current module (highlighted in red) is contributing on Basics of Java Programming.



* **Recommended duration: 24 hours**

Module Objectives

By the end of this module, you will be able to:

- Define Unix Operating System
- Use Unix Internal and External commands
- Create Files and Directories and User Permissions at different level
- Use the Filter commands and Pipes
- Use Input/Output Redirection
- Use Vi editor
- Create and Manage User Accounts
- Start and Kill processes in Unix

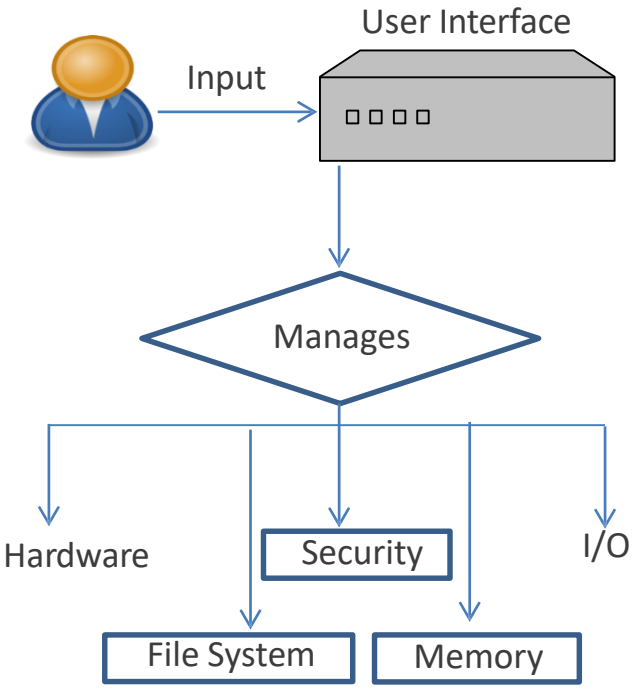
Unix Operating System - Define UNIX Operating System

What is UNIX Operating System?

- The UNIX operating system is a set of programs that acts as an interface between the computer and the user.
- The computer programs that allocate the system resources and coordinate all the details of the computer's internals is called the operating system or kernel.

Why UNIX Operating System?

- Multitasking with protected memory - Multiple users can run multiple programs each at the same time without interfering with each other or crashing the system.
- Access controls and security. All users must be authenticated by a valid account and password to use the system at all times. All files are owned by particular accounts. The owner can decide whether others have access to read or write his files.
- Available on a wide variety of machines - the most truly portable operating system.



Operating System

References

- http://en.wikipedia.org/wiki/Operating_system - Operating System
- http://www.tutorialspoint.com/unix/unix-getting-started .htm - UNIX Operating System

Unix Internal and External commands - Use UNIX Internal and External commands

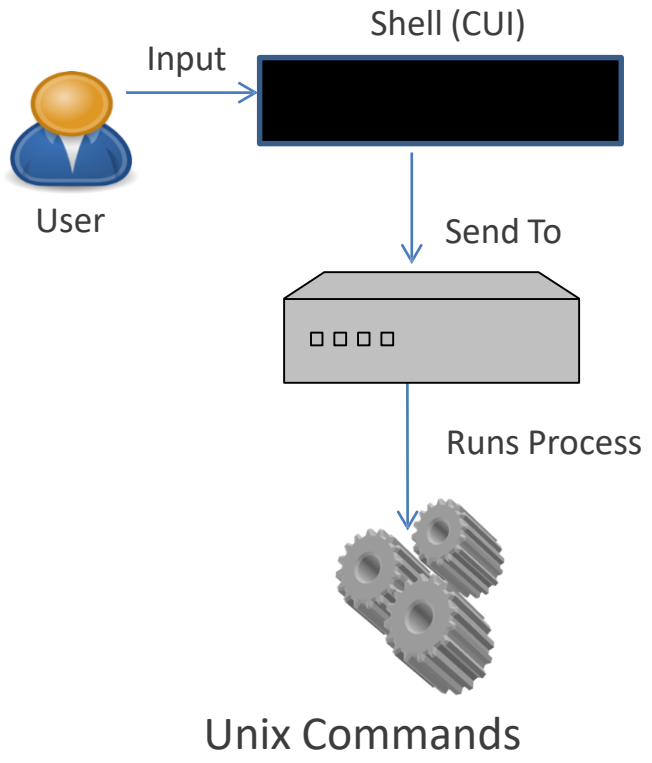
What and why UNIX Internal and External commands?

- The commands are instructions to the operating system.
- All the commands of UNIX are entered in shell (CUI). Shell is a portion of Unix that handles the command liner inputs.

The general syntax for a UNIX command is

$ command -options targets

- UNIX commands can often be grouped together to make even more powerful commands with capabilities known as I/O redirection (< for getting input from a file input and > for outputting to a file) and piping using '|' to feed the output of one command as input to the next.
- For more details of each command type '$ man' command name in Unix shell.



References	
	• http://www.tutorialspoint.com/unix/unix-useful-commands.htm - Useful Commands • http://www.tjhsst.edu/~dhyatt/superap/unixcmd.html - Useful Commands

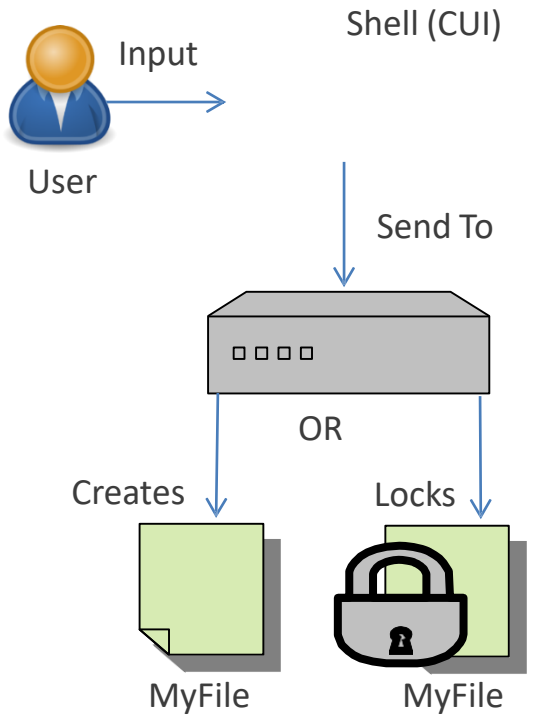
Files and Directories - Create Files and Directories and User Permissions

What are files and directories?

- All data in UNIX is organized into files. All files are organized into directories. These directories are organized into a tree-like structure called the file system.
- When you work with UNIX, one way or another you spend most of your time working with files.

User Permissions:

- File ownership is an important component of UNIX that provides a secure method for storing files. Every file in UNIX has the following attributes
  - **Owner permissions:** The owner's permissions determine what actions the owner of the file can perform on the file.
  - **Group permissions:** The group's permissions determine what actions a user, who is a member of the group that a file belongs to, can perform on the file.
  - **Other (world) permissions:** The permissions for others indicate what action all other users can perform on the file.



Files & File Permissions

References	
	• http://www.tutorialspoint.com/unix/unix-file-management.htm - Files and Directories • http://www.tutorialspoint.com/unix/unix-file-permission.htm - User Permissions

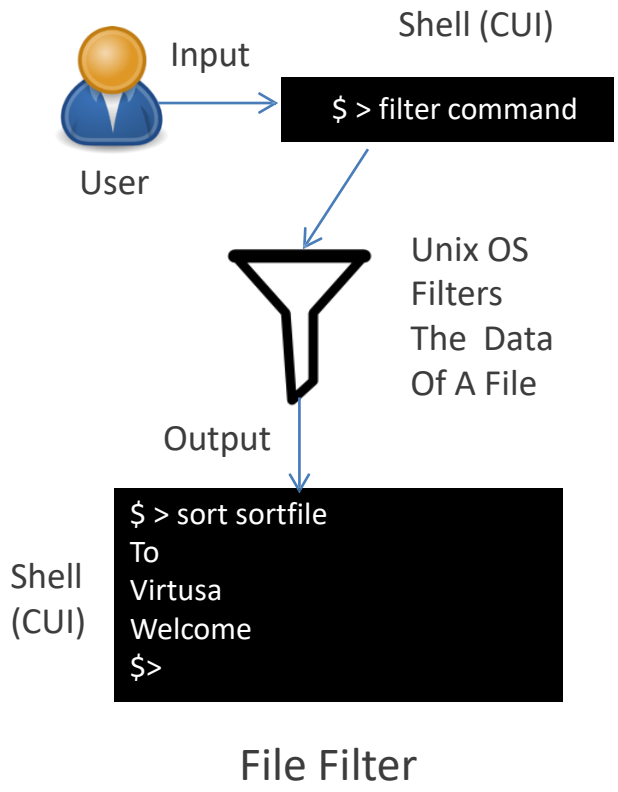
Filter Commands and Pipes - Use the Filter Commands and Pipes

What are Filter Commands and Pipes?

- Using filter commands, a user can perform operations like add/ edit/ search/ sort text in a file or files in directories.
- Many Unix programs read some input, perform a transformation on it, and write it to some output. These programs are called **Filters** and when used together with pipes can produce powerful programs.
- Two or more commands connected in this way form a **Pipe**. You can connect two commands together so that the output from one program becomes the input of the next program.
- To make a pipe, put a vertical bar (|) on the command line between two commands.

Why to use Filter Commands and Pipes?

- Combining the commands together, one can accomplish complex tasks with ease.



File Filter

References	
	• www.csse.monash.edu.au/~unixprac/guest/docserve.epl?doc=doc/unix_filters.html&topic=unix_filters • http://www.tutorialspoint.com/unix/unix-pipes-filters.htm - Filters

Input/Output Redirection - Use Input/Output Redirection

What is I/O redirection?

A command normally reads its input from a place called standard input, which happens to be Key Board by default. Similarly, a command normally writes its output to standard output, which is terminal by default.

Output Redirection:

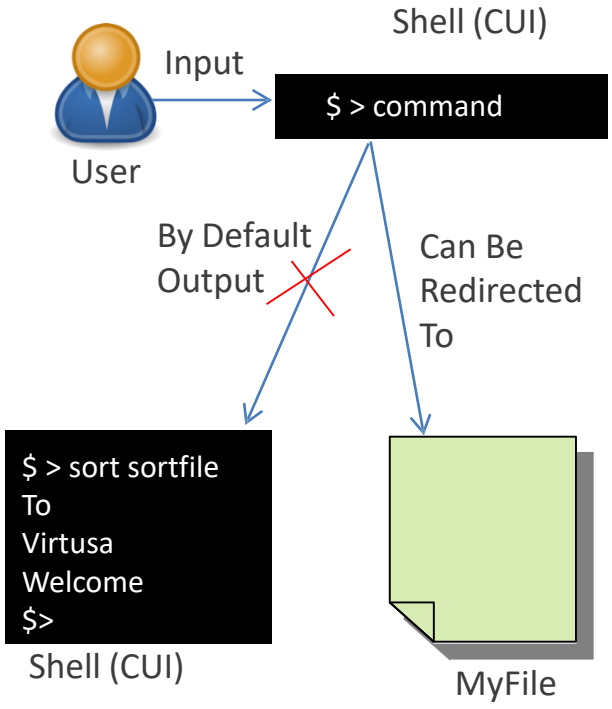
- The output from a command normally intended for standard output, can be easily diverted to some other output destination (Eg: file). This capability is known as output redirection.

Input Redirection:

- The input of a command can be redirected from some other input source.

Why to do I/O redirection?

- To log or store the output in permanent storage mediums for future analysis etc.
- To read the input from different sources to make application more user-friendly etc.



Example I/O Redirection

References	http://teaching.idallen.com/cst8207/12f/notes/200_redirection.html

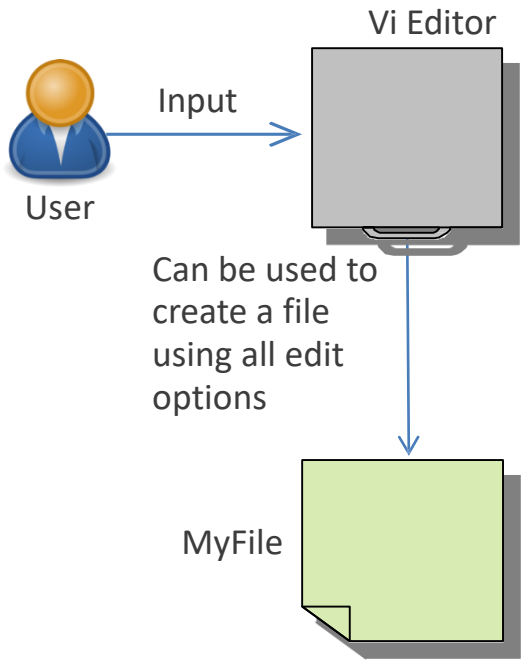
Vi Editor - Use Vi Editor

What is Vi Editor?

- The Vi editor (short for Visual Editor) is a screen editor which is available on almost all Unix systems. Once you have learned Vi, you will find that it is a fast and powerful editor. Vi has no menus but instead uses combinations of keystrokes in order to accomplish commands. There are many ways to edit files in Unix and one of the best ways is using screen-oriented text editor Vi.
- This editor enables user to edit lines in context with other lines in the file.

Why to use Vi editor?

- It is usually available on all the flavors of Unix system.
- Its implementations are very similar across the board.
- It requires very few resources.
- It is more user-friendly than any other editors like ed or ex.
- You can use Vi editor to edit an existing file or to create a new file from scratch. You can also use this editor to just read a text file.



Vi Editor

References	http://www.tutorialspoint.com/unix/unix-vi-editor.htm

Create and Manage User Accounts - Explain Commands to Create and Manage User Accounts

What are different types of User Accounts in UNIX?

- **Root account:** This is also called Super User and would have complete and unfettered control of the system. A super user can run any commands without any restriction. This user should be assumed as a system administrator.
- **System accounts:** System accounts are those needed for the operation of system-specific components for example mail accounts and the sshd accounts. These accounts are usually needed for some specific function on your system, and any modifications to them could adversely affect the system.
- **User accounts:** User accounts provide interactive access to the system for users and groups of users. General users are typically assigned to these accounts and usually have limited access to critical system files and directories.

Why to have different types of User Accounts in UNIX?

- To achieve better Access controls and security, all users must be authenticated by a valid account and password to use the system at all. All files are owned by particular accounts. The owner can decide whether others have read or write access to his files.

Create and Manage Users

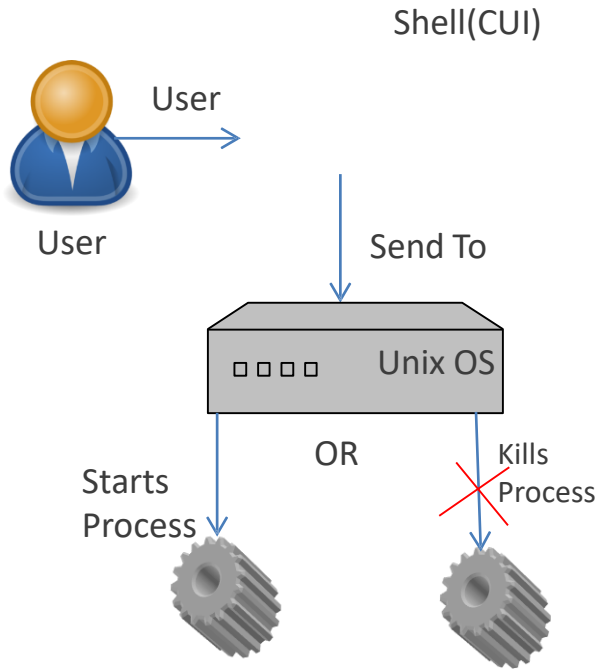
References	http://www.tutorialspoint.com/unix/unix-user-administration.htm

*image source: http://www.industrycortex.com/files/2/68367/508254557.jpg

Start and Kill Processes in Unix - Explain Commands to Start and Kill processes

What is a Process in UNIX?

- A process is executing a program.
- Whenever a program is executed on UNIX system, the system creates a special environment for that program. This environment contains everything needed for the system to run the program as if no other program were running on the system.
- Whenever a command is executed in UNIX, it creates or starts, a new process. For example when 'ls command' is executed, UNIX starts a new process.
- A process, in simple terms, is an instance of a running program.
- The operating system tracks processes through a five digit ID number known as the **pid** or process ID . Each process in the system has a unique **pid.**
- Pids eventually repeat because all the possible numbers are used up and the next pid rolls or starts over. At any one time, no two processes with the same pid exist in the system because it is the pid that UNIX uses to track each process.



Start and Kill Process

References	http://www.tutorialspoint.com/unix/unix-processes.htm

Additional References

The last discussed topics aimed at to gain the concept knowledge and the practical contexts to apply the knowledge.

If you want to explore more and build the expertise level, refer to below links and books:

Links :-

http://www.ee.surrey.ac.uk/Teaching/Unix/
http://www.cs.sfu.ca/~ggbaker/reference/unix/ http://www.sikh-history.com/computers/unix/commands.html

Books :-

- Unix Nutshell
- Unix Complete Reference

Self Check?

Instructions to write Self Evaluation Sheet:

Open the excel sheet, refer OS UNIX sheet, write down the solutions for all questions, save a local copy in your machine.

Assignment

- Refer *Assignment Document* to complete the tasks on the required timeline.
- You are required to submit the Solutions for the given assignment and refer the *Participant guide* to get know the submission procedure.

Module Summary

Now that you have completed this module, you will be able to:

- Define Operating System internals
- Use Unix Internal and External commands to Files and Directories, Users and User Permissions at different level
- Use the Filter commands, Pipes and Input/output Redirection
- Use Processes commands

Thank you!