



Mid-Term Exam: Study Guide

MidTermStudyGuide

Table of Contents

- [1. Command-Line Shells - Bash and Powershell](#)
 - [1.1. Important terms:](#)
 - [1.2. Files and Folders](#)
 - [1.2.1. Wildcards and regular expressions](#)
 - [1.2.2. Access control](#)
 - [1.3. Input and output](#)
 - [1.3.1. Pipelines and filters](#)
 - [1.4. Process and job control](#)
- [2. Text Editors: Vim and Emacs](#)
 - [2.1. Vim](#)
 - [2.2. Emacs](#)
- [3. Programming in Bash](#)
 - [3.1. Conditional Statements](#)
 - [3.2. Looping statements](#)
- [4. Programming in Perl](#)
 - [4.1. Variable and Data Types](#)
 - [4.1.1. Scalars: Numbers and Strings](#)
 - [4.1.2. Arrays](#)
 - [4.1.3. Hashes](#)
 - [4.2. Conditional statements](#)
 - [4.3. Loops](#)
 - [4.4. Files](#)
 - [4.5. Regular expressions](#)
 - [4.5.1. Regular expressions in the shell](#)
 - [4.5.2. Regular expressions in Perl](#)

1 Command-Line Shells - Bash and Powershell

1.1 Important terms:

- Command Processor, Prompt, Command, Parameter, login shell, window manager, remote shell, terminal application, file, folder, directory, pipeli

1.2 Files and Folders

- Folder and directory as synonyms
- The concepts of current directory and working directory
- Design and structure of the file system
 - Difference in layout between the Linux and Windows folder structure
- Difference in behavior between files and folders
- Basic commands for working with the file system: cd, pwd, mkdir, rm, rmdir, ls, find, and grep

1.2.1 Wildcards and regular expressions

- How to use wildcards to represent groups of files
- Use of regular expressions to specify files

1.2.2 Access control

- The concept of access control bits
- Structure of basic Linux access control: user, group, and other
- Differences in interpretation between access control for files and folders
- Use of the chmod, chown, and chgrp commands
- Use cases of file permissions

1.3 Input and output

- The standard system files: stdin, stdout, and stderr
- File redirection: Redirection of standard system files to files in the file system

1.3.1 Pipelines and filters

- Concept of pipeline of mapping output from one command to input of next command
- Concept of filter in the pipeline

1.4 Process and job control

- Concept of process and job
 - Basic commands for viewing currently running processes: ps, top

- Foreground and background applications: fg, bg, jobs
- Common use cases

2 Text Editors: Vim and Emacs

- Important terms: text file, binary file, text editor, regular expression

2.1 Vim

- Basic commands: open, save, exit, navigation by character and word
- Immediate commands: search and replace, regular expression search
- Advanced commands: regular expression search, running commands in editor

2.2 Emacs

- Basic commands: open, save, exit, navigation by character and word
- Immediate commands: search and replace, regular expression search
- Advanced commands: regular expression search, running commands in editor

3 Programming in Bash

- Basics:
 - Naming conventions for shell scripts
 - Difference between running shell scripts through Bash vs. direct execution
 - What is a sha-bang and what does it do?
 - Parameters and variables
 - Important terms: Positional parameter, special parameter, shell variables, environment variable
 - Use of positional parameters to get values from the command-line
 - Special parameters
 - Quoting and Expansion
 - Difference in processing between double quotes and single quotes

3.1 Conditional Statements

- The most irritating thing about shell programming: what is meant by true and false
 - Remember the difference between shells programming and other programming languages
- The Test command
 - Use of the test command in if-statements
 - Shortcut versions of the command and oddities of syntax
 - Test operators: checking for files, checking for type of values, testing regular math, combining conditions
 - List control operators: purpose and use
- Syntax of an if, if-else, and if-else-if statement in the shell
- Syntax of a case statement in the shell

3.2 Looping statements

- Syntax of a while-loop, until-loop
- Use of for-loops in bash
 - Syntax (including the darn semicolon)
 - Range based for-loops

4 Programming in Perl

- What is Perl?
 - The two overriding design guidelines
 - There Is More Than One Way To Do It
 - Easy things should be easy and hard things should be possible
 - Perl as an interpreted language
 - The REPL concept
 - Use of the Perl command line
 - Use of the sha-bang in Perl programs

4.1 Variable and Data Types

- Duck Typing: If it walks like a duck, talks like a duck, and smells like a duck, then there is a gosh-darn good chance it's a duck.
 - Data type inference from first use
- The Three Basic Perl Data Types: Scalar, Arrays, and Hashes
 - Use of decorators on variable names
 - The role of the my keyword
- Data type conversions
 - What are some of the odd things with how data types are converted in Perl?
 - Operator overloading is avoided, means things get interpreted very differently than you expect
 - Note the provision of different operators for comparing numeric and string quantities and what happens when you mix such things

4.1.1 Scalars: Numbers and Strings

- Integers and floats
- Strings
 - Use of the substr function
 - Use of the split function

4.1.2 Arrays

- Initialization of arrays
- Use of scalars to access elements of an array
- Assignment of array to scalar returns length of array

4.1.3 Hashes

- Dictionary data structure: stores key-value pairs
- Use of initializers
- The keys and values functions

4.2 Conditional statements

- Basic syntax of if, if-else, and if-elseif-else constructs
- Use of the unless and unless-else constructs
- Switch statements syntax and their use

4.3 Loops

- Basic syntax of the while and until constructs
- Behavior of range-based loops

4.4 Files

- Difference between print and say
- File operations: open, close
- The readline construct and its relationship to loops

4.5 Regular expressions

- What is a regular expression?
 - Definitions: pattern matching, regular expression
- Basic syntax for working with a regular expression (note: test will include a cheat sheet)

4.5.1 Regular expressions in the shell

- Use of regular expressions when dealing with files and the file system
- Purpose of the grep command in the shell
- The use of the find command and find-grep pipeline

4.5.2 Regular expressions in Perl

- Three classes of regular expressions: match, substitution, and translate
 - Be able to explain the differences
 - Understand the side effects of each type
- The binding operator
 - Use in conditional statements and loops
 - Regular expression variables
 - The before, matched, and after variables
 - Return variables

Date: 2018-06-17 Sun 00:00

Created: 2019-06-13 Thu 16:34

Emacs 25.3.1 (Org mode 8.2.10)