Assignment 2: Shell Scripting

Shell scripting is an important skill for DevOps engineers, software developers, and system administrators. In this assignment, you will create shell scripts to complete two small projects designed to reinforce your skills in shell scripting.

1. **System Setup Scripts**: Your first task is to develop several small scripts that automate the installation of essential software packages and create symbolic links to configuration files stored in a remote Git repository. These scripts will facilitate the initial setup of your system and establish an efficient workflow for managing configuration files.
2. **User Creation Script**: The second script will automate the process of creating a new user on the system. This includes setting a password, establishing preliminary group memberships, creating a home directory, and configuring the user's shell. This task is vital for maintaining user management and ensuring proper access control within a system.

Through these exercises, you will learn how to harness the power of shell scripting to enhance system administration tasks. By the end of this assignment, you will have gained valuable experience in scripting, package management, and user administration—all essential skills for a successful career in system administration.

These assignments will also help develop a better understanding of control flow and generally improve your skills in writing code.

General Instructions and Considerations

As you work on your shell scripts, please adhere to the following guidelines to ensure clarity, functionality, maintainability, and that learning objectives are met:

1. **Documentation**: Write well-documented code. Use comments throughout your scripts to explain the purpose of various sections and any complex logic. Almost every line should have a comment that explains what it does. If you use a resource to find information, include a link to that resource in your comment.
2. **Command-Line Options**: Utilise `getopts` in both scripts to handle command-line options. This allows users to customise the behaviour of your scripts through easily understandable flags. Ensure that your scripts can gracefully handle invalid options and provide helpful feedback.
3. **Error Handling**: Implement robust error handling in your scripts. Check for potential issues (such as missing arguments or failed commands) and return informative error messages to users that guide them in resolving the issues.
4. **Development Environment**: Write your scripts in the Arch Linux environments used in class (the droplets). Use Neovim as your text editor.
5. **Language and Formatting**: The only language that should be used in your scripts is Bash. Ensure your scripts are Unix formatted, not DOS formatted.
6. **Best Practices**:
    - Avoid using `cd` in scripts; instead, use absolute or relative paths.
    - Avoid using `sudo` in a script; require the script to be run with sudo or as root. Include a check for this.
7. **Original Work**: Code must be written by you! Using an AI tool such as ChatGPT or Co-Pilot is prohibited and could result in a score of 0 on the assignment.

Project 1: Configuration Scripts

Setting up a new system can be time-consuming. You have to install packages and copy all of your configuration files into different locations to set up software the way you like it. Scripting is a common solution to this problem. You are going to write scripts to streamline the setup process for new systems.

The configuration scripts will include the following components:

- A user-defined list of packages to be installed
- A script to install those packages
- Existing configuration files for several applications (these have been provided)
- A script to make symbolic links from those configuration files (this is generally done as part of a dotfiles management solution to keep your files organised in a remote repo)
- A script to call the other two scripts as required

Start this script by cloning the example configuration files directory here. This includes the below directory structure:

- bin/
    - sayhi
    - install-fonts
- config/
    - kak/
        - kakrc
    - tmux/
        - tmux.conf
- home/
    - bashrc

To use the configuration in the config directory you need the following packages:

- kakoune
- tmux

For the `bin` and `config` directory your script should preserve the directory structure creating a `~/bin` and `~/.config` that are symbolic links to the original files and directories. For the `home` directory the `bashrc` file should be a symbolic link to `~/.bashrc`

Project 2: New User Script

Setting up a new user is a common task for anyone using a computer. Creating your own script to handle this task will help you better understand the process.

You cannot use an existing tool for creating a new user! Write to the required files directly in your script.

Your script will allow a user to create a new user with the following:

- A specified shell
- A home directory created

- The contents of the `/etc/skel` directory copied into their home directory
- Additional groups that they could be added to (the new user should have a primary group that matches their username)

In addition to the above, the new user should have a password created using the `passwd` utility.

Submission

Submit both of your projects in a public Git repository (GitHub, GitLab, Bitbucket, Codeberg... the choice is yours). Store your projects in separate directories. Include a README.md file that provides a description of your scripts and any additional information needed to use them.

Use the D2L dropbox to submit a link. Double-check that you have a working link! Test it by sending it to a friend or family member if you don't trust yourself to test it.

Grading

Grading Element	Excellent	Good	Satisfactory	Insufficient
Functionality of Scripts	20%	15%	10%	0%
Documentation	15%	10%	5%	0%
Error Handling	15%	10%	5%	0%
Bash scripting understanding	15%	8%	5%	0%
Use of `getopts`	10%	8%	5%	0%
Code Clarity and Comments	10%	8%	5%	0%
Unix Formatting	5%	4%	3%	0%
Use of Git	5%	4%	3%	0%
Overall Submission Quality	5%	4%	3%	0%
Total	**100%**			