

Building Menus in BASH



Andrew Mallett

LINUX AUTHOR AND CONSULTANT

@theurbanpenguin www.theurbanpenguin.com



Module Overview



Creating menus in BASH

Creating CASE statements to evaluate many conditions

Adding user accounts

Removing users

Adding the menu to a login script



Building BASH Menus

Using a BASH menu allows you to present operators with a set list of controlled tasks they are allowed to carry out. Making life more simple and accurate. The menu will loop until users quit the menu



Menu

```
while true ; do
    clear
    echo "Choose 1, 2, or 3"
    echo "1: Calendar"
    echo "2: Date"
    echo "3: Quit"
    read -sn1
    case "$REPLY" in
        1) cal;;
        2) date;;
        3) exit 0;;
    esac
    read -n1 -p "Press any key to continue"
done
```



Demo



Building a simple menu and using case statements



Function to Create Users



As we run from the menu now we do not need to check for parameters



Using the command `chpasswd` makes the script more usable across distributions. The `passwd` command accepts standard input in Red Hat but not Debian based systems



Echo a success message at the end



Demo



Convert the create user code to
a function



Function to Delete Users



We will create a new prompt as we do not need the password



Test for ! `check_user` as we want to error if the user is NOT there



Using `return 1` will exit the function but not the menu if the user does not exist



Demo



Create function to delete user and build the menu



Scripts on Login

Adding the menu script to the `~/.bash_profile` will ensure that it runs on login. Using `exec <path-to-script>` will ensure the user is logged out when they exit the menu



Demo



Add the menu to login script



Summary



Menu execute code within **WHILE** loop

Start with **clear** to erase the screen

End with prompt to allow output to be read

Case statements allow for menu selection

Adding the script to a user's **.bash_profile** automates the script execution

Using **exec** causes a logout on menu closure