Assignment 1: Bash commands and scripts (7%)

SoftEng 206, Semester 2, 2017

Due: Monday 14^{th} August, 1pm

Learning Outcomes

The purpose of this assignment is to target the following learning outcomes, as listed in the SoftEng 206 course outline. On successfully completing this assignment, you will:

Learning Outcome #2: *be comfortable using the command line to perform file processing, writing/running scripts, and various system calls.*

The assignment is also working towards some additional learning outcomes, which you will continue to develop throughout the course:

Learning Outcome #1: *be able to explain the common issues that arise in the construction of software.*

Learning Outcome #3: *be able to explain the purpose of, and how to use, a number of the tools commonly used in the construction of software.*

Assessment

- This assignment is worth 7% of your course mark. Your submission must include the following two files submitted via Canvas. You should use ZIP or TAR to archive your submission! **Do not use any other archive format.**
  - A single bash script file for your submission, and
  - A **signed and dated Cover Sheet** stating that you worked on the assignment independently, and that it is your own work. Include your name, ID number, the date, the course and assignment number. You can generate and download this in Canvas, see the Cover Sheet entry.
- **All programs will be executed on the ECE Remote Linux image**. Be sure to test *all* your work using the command line, as all programs will be executed this way. You will lose a lot of marks if the markers cannot get your program to run on the ECE lab's Linux machines, this is especially important if you developed your code via Cygwin or the likes.

Late submissions

Late submissions incur the following penalties:

- 15% penalty for zero to 24 hours late
- 30% penalty for 25 to 48 hours late
- 100% penalty for over 48 hours late (dropbox automatically closes)

You must double check that you have uploaded the correct code for marking! There will be no exceptions if you accidentally submitted the wrong files, regardless if you can prove you did not modify them since the deadline. No exceptions.

Introduction

This assignment requires that you write a Bash script file that will simulate the authoring component of a voice-activated Mathematics learning aid. Even though you are creating a program that will be executed from the command line, it should take into consideration *ease of use* and *error handling*, for example:

- Check for file names where an existing file already exists (ask the user if the intention is to override).
- For novice users, provide a step-by-step ability to specify the necessary information.
- When incorrect input given, should not quit the program entirely, or abort an operation entirely. Give the user the chance to correct it and continue.
- Because of the interactive nature of the script, think carefully when you print output to the user. Let's say you have just displayed the list of creations; do not immediately print out the menu again, as they might not have had a chance to notice the output as it quickly gets scrolled away. Instead, check that the user has read the output by asking them "Press any key to continue", then proceeding to displaying the menu again.

The Main Menu

When your script file is executed, it should present the user with a menu as follows:

```
> ./maths_aid.sh
============================================================
Welcome to the Maths Authoring Aid
============================================================
Please select from one of the following options:

    (l)ist existing creations
    (p)lay an existing creation
    (d)elete an existing creation
    (c)reate a new creation
    (q)uit authoring tool

Enter a selection [l/p/d/c/q]:
```

The script will continue to prompt the user for an input from one of the above options, until quit is specified. **All video/audio/text files created must be stored and managed neatly in the same subfolder where the `maths_aid.sh` script file is located**. Do not generate any files anywhere else.

What is a "creation"?

The purpose of this application is to author/create videos, known as "creations", that would be helpful in teaching beginners to recognise and say numbers:

- A creation would be for a given number.
- For the video component, it displays that number as ordinary text in the video, e.g. "nine" or "9".
- For the audio component, it sounds the number. The sound will come from the user recording themselves saying that number.
- The video and audio components are merged together to form the creation.

(l)ist existing creations

This menu option will list all the current creations that the user has created. It is important that you list these tidily. You should list the creation name (see last feature in "create"), not the filename (i.e. don't display filename extensions or file paths). Any display format is fine, as long as it clearly shows the user how many creations there are (without the user having to count them), it is tidy, and so on. Think carefully also what happens when there are no creations (e.g. none were created yet, or all were deleted).

(p)lay an existing creation

Here, the user would like to watch a particular creation, assuming at least one exists. You should remind the user what existing creations exist (by listing them again), and allow them to select one to play. As soon as the creation is played, the video should automatically close and show the menu again. Think carefully how you program this part. You don't want to copy-paste the same functionality as the list option. Think in terms of code reuse.

(d)elete an existing creation

Very similar to the play functionality above, except you delete the creation the user specifies. You therefore initially show them the list of creations they can select from, and once they select one to delete, you should confirm with them if they are sure they want to delete it (make sure you spell out what they are attempting to delete). Again, think in terms of code reuse when displaying the existing creations.

(c)reate a new creation

This is the main part of the authoring aid, where the user will generate a new creation. The process is as follows:

1. Ask the user to enter a name for the new creation. You need to make sure that there isn't an existing creation with that same name. If there is already one with that name, you need to keep asking the user to enter another name until they enter one that doesn't already exist.
2. You generate the video part of the creation, where the name forms the text content of the video. It doesn't matter how the text is displayed, as long as it's clearly visible in the video. The video should be 3 seconds long also.
3. Tell the user that they need to record the audio for the creation. To make sure they are ready, you should have a prompt along the lines of "Press any key to start recording". As soon as they enter any key, it starts recording their voice for 3 seconds.
4. You should offer the user the opportunity to hear the recorded audio, and ask them if they want to `(k)eep` it or `(r)edo` the audio. It the user wants to redo the recording, you repeat step 3 above. This is repeated until the user is happy with the audio recording.
5. You then combine the video (from step 2) with the audio (from step 4) into a single video, and this is your creation.

(q)uit authoring tool

Every time the user performs an operation, they should be presented again with the main menu (printed again). This should continue indefinitely until the user selects to `quit`.

ffplay and ffmpeg

You will be using the ffplay and ffmpeg commands for playing videos, generating the text videos, recording the audio, and combining the audio/video together for the final creation. **Important notes:**

- Make sure that any printing from other commands (e.g. ffmpeg, ffplay) are not displayed to the screen. You should be redirecting them (stderr and stdout) so as to keep the shell tidy with only what you wrote.
- Remember that your code needs to run in the labs, so you should thoroughly test it (e.g. the location of the fonts might be different in the labs compared to your own laptop).

Resources

You will be introduced to many of the commands to use to get you started. But you will likely have to do some of your own research and find some resources yourself. Here's a start:
`http://tldp.org/LDP/Bash-Beginners-Guide/html/`