Basic Scripting for AV Preservation

Peter Bubestinger-Steindl
(p.bubestinger@av-rd.com)

2020-05

Session 1

- SSH
- Editor (Atom)
- CLI basics review
  - DOS
- Batch 1-0-1 "Edit, save, run. edit…"
- hello_world.bat
- Variables
  - Environment variables
- Arguments (parameters)
- Conditionals
  - Different FFmpeg recipes selected by argument
- Subroutines (GOTO)

Homework

- Write Batchfile to apply a single FFmpeg recipe to a file (given as parameter)
- Try to read and understand DVA "start.bat": https://sourceforge.net/p/dva-profession/code/HEAD/tree/trunk/misc/cli

Session 2

- Programming Dos and Donts
- Loops
  - Process all files in folder with FFmpeg recipe
- Remove suffix `echo %%~nf`

Homework

Session 3

- Install Cygwin
- Shell / BASH
  - CLI basics review
  - Translate previous Batch commands/examples to Shell
  - hello_world.sh

- Functions (vs GOTOs) Example? TODO!
- Variables: Global vs local Example? TODO!
- Tests / conditionals 2
  - Is a file/folder?
  - Is lower/greater than?
- Exit status: First in BASH - then refer back to Batch Example: Run FFmpeg and check if it executed properly. Show error message if not.

Examples: * Renumber DPX file sequence * Arguments: source folder, start index * Start index must be greater-than 0. * Short intro to "printf" and masks

Homework?

Session 4

Homework?

Session 5

- Python Intro

Homework?

Session 6

- Python review
- Summary and Conclusion

–>

Introduction

- SSH: A Secure SHell
- Editor: Atom
- CLI basics review
- Script vs Program?

CLI basics review

- cd
- ls / dir
- rm
- cls
- exit

First Words

- "Hello world!"

First Words

```
echo Hello world!
```

First Words

```
    @echo off
    cls
    echo Hello world!
    @pause
```

Magic Spells 1-0-1

- Variables
- Arguments
- Tests & Conditions
- Exit status
- Loops
- Pipes & Redirection

Programming Dos & Donts

1. Pretty please.
2. Echo loud and clear.
3. Don't trust user input ;)
4. Document it!

Pretty please:

- Proper naming (vars, etc)
- Indent!
- Linebreaks.
- Quotes for strings.
- Be consistent.

Echo loud and clear:

- loglevel?
- silent
- verbose
- debug

Short and superficial introduction to how to echo what's happening and what "loglevel" or "verbosity" means.

Don't trust user input ;)

- Check it.
- Escape it.
- Or forge(t) it.

Check input:

- Does file exist?
- Is this a folder?
- Does it contain illegal characters?
- Is it empty?
- etc.

Escape input:

- Disarm special characters.
- "Escape character"
- Often backslash ""

Forge / forget it:

- Masks and placeholders
- Concatenate values/strings
- If input doesn't make sense:
  Forget it. Bail bout.
  But let the user know "*why*".

Example: Create a string/parameter by concatenating user input in a way that reduces security/error options.

Such as: User input: "ffv1" instead of "-c:v ffv1"

Has pros/cons of course, but we're still learning the first steps here :)

Document it!

- You think you'll remember?
- Clear to you = clear to others?

Clear?

```
i=1
for f in *.dpx; do
    mv $f $(printf %07d $i).dpx
    i=$((i+1))
done
```

Clear.

```
# Set start number of output files:
i=1

# Iterate through all DPX image files:
for FILE in *.dpx; do
    # zero-pad index to 7 digits:
    INDEX=$(printf "%07d" $i)
    mv $FILE $INDEX.dpx
    # Increment counter:
    i=$((i+1))
done
```

Batch (.bat)

- Microsoft DOS/Windows-only
- Basic commands
- Variables / arguments
- Loops
- Conditionals / Goto

I think loops are more interesting/useful here than conditionals, because they are more likely to be needed earlier - and conditionals in batch suck.

Links (.bat)

- Windows Commands (Microsoft Docs)
- A-Z index of commands (ss64.com)
- COMMAND.COM (Wikipedia)
- Tutorial 1 (tutorialspoint.com)
- Tutorial 2 (TryToProgram.com)
- Tutorial 3 (Steve Jansen)
- Errorlevels (Rob van der Woude)
- Batch: Remove file extension (stackoverflow.com)

Bash (.sh)