

Homework 2

Shell Script

huytseng, wuph0612, jschou

國立陽明交通大學資工系資訊中心

Computer Center of Department of Computer Science, NYCU

Usage

- Please provide a executable shell script with following available options:

```
$ hw2.sh -a  
hw2.sh -i INPUT -o OUTPUT [-c csv|tsv] [-j]
```

Available Options:

```
-i: Input file to be decoded  
-o: Output directory  
-c csv|tsv: Output files.[ct]sv  
-j: Output info.json
```

Error Return Code

- Invalid arguments should be rejected with a non-zero status code, with the exact help message outputted to **stderr**.

```
$ hw2.sh -a  
hw2.sh -i INPUT -o OUTPUT [-c csv|tsv] [-j]
```

Available Options:

```
-i: Input file to be decoded  
-o: Output directory  
-c csv|tsv: Output files.[ct]sv  
-j: Output info.json
```

Input File Specification - Introduction

- The to-be-decoded file should have a extension **.hw2** which is actually a JSON file.

Input File Specification - Definition

```
{
  "name": "example.hw2",
  "author": "huytseng",
  "date": 1662786784,
  "files": [
    {
      "name": "example.txt",
      "type": "file",
      "data":
"TG9yZW0gaXBzdW0gZG9sb3Igc2l0IGFtZXQsIGNvbniY3RldHVyIGFkaXBpc2NpbmcgZWxpdC4gTWF1cm1zIGVsZWlmZW5kIHN1bSBsZW8sIGlkIHRpbmNpZHVudCBsb3JlbSB2ZW51bmF0aXMgZWdldC4=",
      "hash": {
        "md5": "9ce543926bc4f3d67368b58e61fb7710",
        "sha-1": "f1d2e348391c502880eb246184f542556005874b"
      }
    }
  ]
}
```

example.hw2

Input File Specification - Definition

```
{
  "name": "example.hw2",
  "author": "huytseng",
  "date": 1662786784,
  "files": [
    {
      "name": "examp
      "type": "file"
      "data":
"TG9yZW0gaXBzdW0gZG9sb3Igc
kIHN1bSBsZW8sIGlkIHRpbmNpZ
      "hash": {
        "md5": "9c
        "sha-1": "
      }
    }
  ]
}
```

example.hw2

- Note:
 - `files.length` ≥ 0 .
 - `files[*].name` will always not exact equal to either **files.csv** or **info.json** since they are reserved.
 - `files[*].type` is always "file".
 - `files[*].name` may imply a non-flatten folder structure. In this case, the required folder should be made. See `generated.hw2` in the Sample Input.
 - The `.hw2` files are not guaranteed to be pretty-printed.

Expected Output files - Structure

- You should write the required files to the designated output directory as output.
- The following file tree is expected if **example.hw2** is the input file.

```
.
├── outputDir/
│   ├── example.txt
│   ├── files.csv
│   └── info.json
```

Expected Output files - Contents

```
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris eleifend sem leo, id  
tincidunt lorem venenatis eget.
```

```
example.txt
```

```
filename,size,md5,sha1  
example.txt,116,9ce543926bc4f3d67368b58e61fb7710,f1d2e348391c502880eb246184f542556005874b
```

```
files.csv
```

Expected Output files - Contents

```
filename    size    md5    sha1
example.txt  116    9ce543926bc4f3d67368b58e61fb7710
f1d2e348391c502880eb246184f542556005874b
```

files.tsv

```
{
  "name": "example.hw2",
  "author": "huytseng",
  "date": "2022-09-10T13:13:04+08:00"
}
```

info.json

Requirements

- Please place your script at `/home/judge/hw2.sh`, with executable bit set.
- The script should start with a proper shebang (`#!/bin/sh`, other shells are **not** allowed.)
- Make sure `/tmp` is writable for the **judge** user.
- SFTP support for the SSH server is required.
- Your script should return the count of *Invalid Files* (which **any** of its checksum mismatches).

Restrictions

- Must not call network tools (such as curl, wget...)
- Must not use any other interpreters, compilers or programming languages (such as Python, Ruby, Node.js, Golang, Rust, Perl, GCC, Clang...)
- Must not call any other self-written scripts, binaries or executables.
- Only one shell, `sh`, is allowed.
- Common tools (e.g. `date`, `base64`, `jq`, etc.) are allowed.
- If you are not sure whether a tool is allowed, please ask TA on Google group.

Grading

Automated grading (Online Judge), 106 pts.

- Usage
 - Invalid options
 - Exit Code (3%)
 - Help Message (4%)
 - Invalid files (8%)
- Input and Output files
 - Arbitrary argument position (15%)
 - Extract single file (15%)
 - Extract multiple files (20%)
 - Output nested directories (10%)
 - Extract & Output **file.csv** (10%)
 - Extract & Output **info.json** (15%)
 - Extract & Output **file.tsv** (6%)

HW 2: Sample input

<https://reurl.cc/V1OevY>

SH(1): getopt

getopts *optstring var*

The POSIX **getopts** command. The `getopts` command deprecates the older `getopt(1)` command.

The first argument should be a series of letters, each possibly followed by a colon which indicates that the option takes an argument. The specified variable is set to the parsed option. The index of the next argument is placed into the shell variable `OPTIND`. If an option takes an argument, it is placed into the shell variable `OPTARG`.

Attention!

- You are restricted to use only **sh** to complete your work
 - That is, no other shell and no other programming language.
 - If you're not sure what's allowed, contact TAs.
 - TAs reserve the right of final explanations. Specs and the points of each sub-judges are subject to change in any time.
- Start from: **10/6 19:00**
- Due date: **10/26 23:59**



Help me! TA!

- Questions about this homework
 - Ask questions on <https://groups.google.com/g/nctunasa>
 - We MIGHT give out hints on google group
 - Be sure to join the group :D
 - Do not email us directly
 - Do not use E3 to email us
- How To Ask Questions The Smart Way
 - <https://github.com/ryanhawu/How-To-Ask-Questions-The-Smart-Way>

Good Luck!

國立陽明交通大學資工系資訊中心

Computer Center of Department of Computer Science, NYCU