

Eighth Session

Joshua Ng • Archives New Zealand
Reto Kromer • AV Preservation by reto.ch

Bash Scripting for Audio-Visual Preservation
webinar series hosted by AMIA
17 February 2022

1

Contents

- globbing
- pattern matching
- regular expressions

2

ASCII (1977/1986)																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1x	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2x	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

Legend:
Yellow: Changed or added in 1963 version
Pink: Changed in both 1963 version and 1965 draft

Wikipedia

3

```
GNU nano 6.1 script_1.sh
#!/usr/bin/env bash

# Bash Scripting for AVpres
# First Session, course version
# 2022-01-25

input="$1"

if [[ "$input" == "Bash" ]] || [[ "$input" = "bash" ]]; then
    bash --version | grep "bash"
elif [[ "$input" == [Ff][Ff]mpeg ]]; then
    ffmpeg -version | grep "FFmpeg"
else
    echo "Usage:"
    echo "  print_version.sh Bash"
    echo "  print_version.sh FFmpeg"
fi
```

Legend:
Yellow: Changed or added in 1963 version
Pink: Changed in both 1963 version and 1965 draft

4

```

GNU nano 6.1 script_5.sh

# generate the checksum of the file and exit with 0
make_checksum() {
  local in_file="${1}"
  local out_file="${2}"
  local algo="${3,,}" # needs Bash 4.0 or newer

  echo -e "${BLUE}Please wait...${NC}"
  ${algo}sum "${in_file}" | sed "s#${in_file%/*}/##" > "${out_file}"
  if (( $? != 0 )); then
    prompt_error "The checksum manifest could not be generated."
  else
    echo -e "${BLUE}... done.${NC}"
    exit 0
  fi
}

# parse the provided input
(( $# == 0 )) && prompt_mini_help
while getopts ":i:o:a:h" opt; do

```

[Read 145 lines]

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line

5

```

GNU nano 6.1 script_6.sh

# verify output file
verify_output() {
  local out_file="${1}"

  if [[ "${out_file}" == '' ]]; then
    prompt_error "No output file passed."
  elif echo "${out_file}##*" | grep -E "^(mov|mp4)$"; (( $? != 0 )); then
    prompt_error "'${out_file}##*' is not a valid file format."
  elif echo '' > "${out_file}_tmp"; (( $? != 0 )); then
    prompt_error "Cannot create an output file '${out_file}'."
  else
    rm "${out_file}_tmp"
  fi
}

return 0
}

# verify file quality

```

[Read 146 lines]

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line

6

Wildchars

?	any single character
*	any string of characters
[set]	any character in set
[^set]	any character not in set
[A-Za-z]	any single character in the range A-Z or a-z

7

Metacharacters

^	start of line or string
\$	end of line or string
(...)	capturing and grouping
... ...	alternation
*, +, ?	greedy quantifiers

8

Quantifiers

...?	zero or one
...*	zero or more
...+	one or more
...{n}	exactly n times
...{n,m}	between n and m times

9

Data filtering

commands

- cat
- grep
- sort
- tr
- curl

languages

- sed
- awk
- Perl
- Python
- PHP
- Java

10

Bibliography

- Jeffrey E.F. Friedl: **Mastering Regular Expressions**, 3rd Edition, O'Reilly, 2006

11

Resources

Rubular

“a Ruby regular expression editor”

→ <https://rubular.com>

12

BagIt

13

BagIt structure

```
data/<archive_content>  
bagit.txt  
manifest-<algorithm>.txt  
bag-info.txt  
tagmanifest-<algorithm>.txt  
fetch.txt
```

14

data/<archive_content>

```
data/element_1  
...  
data/element_n
```

15

bagit.txt

```
BagIt-Version: 1.0  
Tag-File-Character-Encoding: UTF-8
```

16

manifest-sha512.txt

bf29337956dc9398dab716f819e825021db9
78e3d9f58a5697df5ed9cc42074f85686fe8
8e8b097576c21d971948fe5f84123d76ee58
298de7a5d533fe431bf7 data/element_1

...

316a24a5c7b94940dbc430aecb50da996501
bb3bb3a2dd4d6c2c4fcb11ead903ab84f12e
4bdba66f2833f9a6169a7629d3c17a7072c0
fdc6273949ed02d0435b data/element_n

17

bag-info.txt

Bagging-Date: **2022-02-16**

Bag-Size: 84 MB

Payload-Oxum: **87874558.245**

18

tagmanifest-sha512.txt

01d265752482a3044a554c4b8cc85271dab2d124f
a1f2d6b55a334d755d9b61058c4e704a16b44c7d4
eaa43430411744e174d4cd6db282daea493b4f178
7be84 **bag-info.txt**

1d73ae108d4109b61f56698a5e19ee1f8947bdf89
40bbce6adbe5e0940c2363caace6a547b4f1b3ec6
a4fd2b7fa845e9cb9d28823bc72c59971718bb26f
2fbd8 **bagit.txt**

6c83c8fef3979cf057bb06dd167c675a6341b3355
db50938bdb42d979a2c1245a451144c23f482674d
b53230535e50b53cce740263a3e05abe2ac2bbcb0
5a1cf **manifest-sha512.txt**

19

AV Preservation by
reto.ch

zone industrielle Le Trési 3
1028 Préverenges
Switzerland

Web: reto.ch
Twitter: [@retoch](https://twitter.com/retoch)
Email: info@reto.ch



20