

MISE À NIVEAU UNIX : COMPLÉMENTS

ALEXIS BRASY (ALEXIS.BRASY@UVSQ.FR)

ISTY - UVSQ

25 AOÛT 2021

UNIX : GESTION DU TEMPS

Définition

Nombre de secondes écoulées depuis le 1er janvier 1970.

Avantage : manipulation d'un *timestamp* (ie. d'un entier) au lieu d'un format de date complexe.

MANIPULATION DE DATES : date

```
$ date  
Sun 27 Sep 13:42:37 CEST 2020  
$ date +%s  
1601207095  
$ date -d @1601207095  
Sun 27 Sep 13:44:55 CEST 2020
```

STATUT DES FICHIERS

STATUT GÉRÉ PAR LE SYSTÈME DE FICHIERS

Dates :

- d'accès
- de modification
- de changement (eg. modification ou changement des permissions).

```
$ stat a
```

```
File: a
```

```
Size: 49
```

```
Blocks: 8
```

```
IO Block: 4096
```

```
Device: fd01h/64769d Inode: 49414301 Links: 1
```

```
Access: (0664/-rw-rw-r-) Uid: ( 1000/ alexis) Gi
```

```
Access: 2020-09-27 14:03:39.475278510 +0200
```

```
Modify: 2020-09-27 14:03:39.475278510 +0200
```

```
Change: 2020-09-27 14:03:39.475278510 +0200
```

```
Birth: -
```

Dates :

- d'accès (avec touch -a)
- de modification (avec touch -m)
- de changement (géré par l'OS, ne peut pas être manipulé).

ANALYSE STATIQUE

QU'EST-CE QUE *L'ANALYSE STATIQUE*

Définition ¹

En informatique, la notion d'analyse statique de programmes couvre une variété de méthodes utilisées pour obtenir des informations sur le comportement d'un programme lors de son exécution sans réellement l'exécuter.

Pour les scripts shell :

- shellcheck
- <https://www.shellcheck.net>.

1. https://fr.wikipedia.org/wiki/Analyse_statique_de_programmes

EXAMPLE

```
$ cat script
#!/bin/bash
a=$(ls)
echo a
$ shellcheck script
```

In script line 2:

```
a=$(ls)
^-- SC2034: a appears unused. Verify it or export it.
```

EXAMPLE

```
$ cat script
#!/bin/bash
a=$(ls)
echo $a
$ shellcheck script
```

In script line 3:

```
echo $a
    ^- SC2086: Double quote to prevent globbing and
```

EXAMPLE

```
$ cat script
#!/bin/bash
a=$(ls)
echo "$a"
$ shellcheck script
```

ÉCRITURE FLUX STANDARDS DANS UN SCRIPT

ÉCRITURE FLUX STANDARDS DANS UN SCRIPT : EXEMPLE

```
$ cat script
echo "standard output" > /dev/stdout
echo "error output" > /dev/stderr
$ ./script >out 2>err
$ cat out
standard output
$ cat err
error output
```

QUESTIONS?