

M1101-3 Initiation Linux

Mémento Syntaxique

Ne pas écrire sur ce document

(seul document autorisé au DS de promo)

Année 2015-2016

cut [*<option(s)>*] [file...]

Options :

<i>valeur</i>	<i>action</i>
-d <x>	force le caractère x comme séparateur. Par défaut, le séparateur est l'espace,
-c <liste>	extraît tous les caractères dont le rang est désigné par liste,
-f <liste>	extraît tous les champs dont le rang est désigné par liste. liste peut être égale à : m-n ou m,n ou n

egrep [*<option(s)>*] pattern [file ...]

Options :

<i>valeur</i>	<i>action</i>
-v	affiche toutes les lignes ne contenant pas la chaîne,
-c	affiche le nombre de lignes contenant la chaîne,
-l	affiche les noms des fichiers qui contiennent la chaîne,
-n	affiche numéro de ligne devant chaque ligne contenant la chaîne,
-i	les majuscules et minuscules sont confondues pour la recherche.

ps [*<option(s)>*]

Options :

<i>valeur</i>	<i>action</i>
-e	affiche tous les processus,
-f	permet un affichage détaillé (propriétaire, PPID...)
-H	affiche la hiérarchie des processus.

sed commande [file...]

Commande :

s/pattern/string/ substitue la 1^{ère} occurrence du pattern par string (une par ligne),
s/pattern/string/g substitue toutes les occurrences de pattern par string.

sort [*<option(s)>*] [file]

Options :

<i>valeur</i>	<i>action</i>
-b	ignore espaces et tabulations en en-tête,
-d	seuls les lettres, les chiffres et les espaces sont évalués,
-f	les majuscules et minuscules sont confondues pour le tri,
-r	ordre inverse,
-n	effectue un tri numérique,
-t <x>	considère x comme le séparateur de champs,
-k <n>	effectue le tri sur le nième champs,

MANUEL SIMPLIFIE DE COMMANDES

- + <m.n> désigne la position de départ de la clé du tri; elle commence au (n+1)ème caractère du (m+1)ème champ. Les champs sont séparés par une tabulation (sauf si l'option **-t** est employée),
- <m.n> désigne la position de fin de la clé du tri.
- o <fichier> fichier de sortie

tar [<option(s)>] -f archive-file [files | directories]

Options :

- | <i>valeur</i> | <i>action</i> |
|---------------|-----------------------------------|
| -c | crée une archive, |
| -x | extraie le contenu d'une archive, |
| -t | affiche le contenu de l'archive, |
| -z | l'archive sera compressée |

tr [<option(s)>] string1 [string2] [< file]

Par défaut, le 1^{er} caractère de chaîne1 est remplacé par le 1^{er} de chaîne2, le 2nd par le 2nd, ...

Options :

- | <i>valeur</i> | <i>action</i> |
|---------------|--|
| -d | supprime les caractères de chaîne1, |
| -s | substitue les occurrences multiples de la chaîne par une seule, |
| -c | ce sont les caractères non spécifiés dans chaîne1 qui seront concernés |

wc [<option(s)>] [file...]

Options :

- | <i>valeur</i> | <i>action</i> |
|---------------|---|
| -l | affiche le nombre de lignes du fichier, |
| -w | affiche le nombre de mots du fichier, |
| -c | affiche le nombre de caractères du fichier. |

Par défaut, la commande affiche le nombre de lignes, de mots et de caractères du fichier.

Métacaractères du shell

*	une chaîne quelconque
?	un caractère quelconque
[...]	caractères possibles
[^...]	caractères interdits

Métacaractères dans les expressions régulières

.	un caractère quelconque
[...]	caractères possibles
[^...]	caractères interdits
^	début de ligne
\$	fin de ligne
?	zéro à une occurrence du caractère qui précède
*	zéro à n occurrence(s) du caractère qui précède
+	une à n occurrence(s) du caractère qui précède
{n}	n occurrences du caractère qui précède
{n, m}	n à m occurrences du caractère qui précède
(e1 e2)	expression e1 ou expression e2
\	échappement du caractère qui succède

Tests

test *expression* ou ! test *expression* avec *expression* :

Entiers et Chaînes

\$s1 = \$s2	les variables s1 et s2 ont des contenus identiques
\$s1 != "toto"	la variable s1 a un contenu différent de « toto »

Variables

-z \$b	la variable b est vide
--------	------------------------

Entiers

\$n1 -gt \$n2	le contenu de la variable n1 est plus grand que celui de n2
\$n1 -ge 2	le contenu de n1 est plus grand ou égal à 2
\$n1 -lt \$n2	le contenu de n1 est plus petit que celui de n2
\$n1 -le \$n2	le contenu de n1 est plus petit ou égal à celui de n2

Fichiers

-e f1	le fichier f1 existe
-s f1	le fichier f1 est non vide
-d f1	le fichier f1 est un répertoire

METACARACTERES ET TESTS

-f f1	le fichier f1 est un fichier ordinaire
-r f1	le fichier f1 est lisible
-x f1	le fichier f1 est exécutable
-w f1	le fichier f1 est modifiable

Exemple d'expression booléenne complexe :

`(test $a -gt $b || test $a -gt $c) && test $d -gt $a`

Les structures de test

<pre> if ... then ... elif ... then ... else ... fi </pre>	<pre> case <i>variable</i> in <i>motif1</i>) ... ;; <i>motif2</i> <i>motif3</i>) ... ;; *) ... esac </pre>
--	--

Les boucles

<pre> while ... do ... done </pre>	<pre> until ... do ... done </pre>	<pre> for <i>variable</i> in ... do ... done </pre>
--------------------------------------	--------------------------------------	---