



Étape : Licence Sciences & Technologie - Informatique

Épreuve de : Utilisation des systèmes informatiques

Date : 10 mai 2007

Heure : 08h30

Documents manuscrits et livres autorisés

Épreuve de Monsieur Delmas

UE : USI - INF103

Durée : 1h30

Répondre dans les cadres directement sur le sujet ou cocher lorsque nécessaire la ou les bonnes réponses.

ATTENTION : cocher une mauvaise réponse retire un point.

_____ *Ce sujet comporte 8 pages sans les annexes et 20 pages avec les annexes* _____

Indiquer votre identificateur d'anonymat :

★ Cochez les systèmes d'exploitation qui sont du type Unix.

Linux	<input type="checkbox"/>	NetBSD	<input type="checkbox"/>
Windows NT	<input type="checkbox"/>	MacOS X	<input type="checkbox"/>
Sun/Solaris	<input type="checkbox"/>	FreeBSD	<input type="checkbox"/>

★ Sous les systèmes du types Unix, quel est le login d'usage du super utilisateur ?

obiwan kenobi	<input type="checkbox"/>	root	<input type="checkbox"/>
superuser	<input type="checkbox"/>	admin	<input type="checkbox"/>

★ La *netiquette* est

l'adresse d'un utilisateur à qui envoyer un courriel <input type="checkbox"/>	l'ensemble des règles de savoir-vivre sur l'Internet <input type="checkbox"/>
l'identificateur du réseau relié à votre machine ... <input type="checkbox"/>	l'adresse d'une page Web

★ Relier par une flèche les cases (applications) de gauches en rapport avec celles de droites (protocoles).

Navigation Web	<input type="checkbox"/>	<input type="checkbox"/>	protocole NNTP
Forums de discussion	<input type="checkbox"/>	<input type="checkbox"/>	protocole NTP
Ordinateur mis à l'heure via un serveur de temps .	<input type="checkbox"/>	<input type="checkbox"/>	protocole SSH
Connexion distante à une machine	<input type="checkbox"/>	<input type="checkbox"/>	protocole SMTP
Courrier électronique	<input type="checkbox"/>	<input type="checkbox"/>	protocole FTP
Récupération/dépôt de documents à distance	<input type="checkbox"/>	<input type="checkbox"/>	protocole HTTP

★ Parmi ces adresses, cocher celle qui n'est pas du type IPv4.

216.239.53.9	<input type="checkbox"/>
198.41.0.4	<input type="checkbox"/>
176.258.122.110	<input type="checkbox"/>
61.213.147.96	<input type="checkbox"/>

★ Quel est le nombre maximum d'adresses IPv4 disponibles ?

2^{16}	<input type="checkbox"/>
2^{32}	<input type="checkbox"/>
2^{64}	<input type="checkbox"/>
2^{128}	<input type="checkbox"/>

★ Comment peut-on passer un processus en avant-plan dans l'état STOP ?

Control-z	<input type="checkbox"/>	Escape	<input type="checkbox"/>
Control-c	<input type="checkbox"/>	Control-d	<input type="checkbox"/>

* Quel est le nombre maximum d'adresses IPv6 disponibles ?

- | | | | |
|-----------------|--------------------------|--------------------------|---------------|
| 2^{16} | <input type="checkbox"/> | <input type="checkbox"/> | 2^{256} |
| 2^{32} | <input type="checkbox"/> | <input type="checkbox"/> | 2^{512} |
| 2^{64} | <input type="checkbox"/> | <input type="checkbox"/> | 2^{1024} |
| 2^{128} | <input type="checkbox"/> | <input type="checkbox"/> | aucune limite |

* Sur les systèmes de type Unix cocher la commande qui permet de changer

- | les droits d'accès : | le groupe Unix d'un fichier : | le groupe Unix de l'utilisateur : |
|--------------------------------------|--------------------------------------|--------------------------------------|
| chgrp <input type="checkbox"/> | chgrp <input type="checkbox"/> | chgrp <input type="checkbox"/> |
| chmod <input type="checkbox"/> | chmod <input type="checkbox"/> | chmod <input type="checkbox"/> |
| chown <input type="checkbox"/> | chown <input type="checkbox"/> | chown <input type="checkbox"/> |

* cocher les affirmations qui sont vraie

- Un utilisateur peut appartenir à plusieurs groupes Unix
- À un instant donné, un utilisateur peut se placer dans un et un seul groupe Unix
- Un fichier à un instant donné appartient à un groupe Unix
- Un fichier à un instant donné peut appartenir à plusieurs groupes Unix
- Les répertoires n'appartiennent à aucun groupe Unix particulier

* Tous les processus se voient attribué un PID unique par le noyau, il n'y a aucune exception :

- Vrai Faux

* Quel est le PID de l'ordonnanceur ?

- 1 1
Cela peut être n'importe quel PID Aucun

* Supposons que vous exécutiez un programme contenant une instruction du type "boucle infinie" (e.g., for(;;) ou while(1)) ce processus va t'il monopoliser de manière continue et indéfiniment le processeur (en d'autres termes, les autres processus sont-ils bloqués ?)

- Oui Non

* Justifier votre réponse précédente

* Rappelez brièvement le principe de fonctionnement d'un "appel système"

- * Si on insère des morceaux de code du noyaux écrits en langage machine dans une application, expliquer pourquoi et comment cette application ne pourra pas se substituer au noyau.

- * Qu'effectuera la commande `bg` dans le contexte suivant ?

```
$ jobs
[1] Stopped          gwenview
[2]- Stopped         firefox
[3]+ Stopped         ps | grep bash
[4] Running          xeyes &
$ bg
```

- | | | | |
|--|--------------------------|---|--------------------------|
| gwenview sera exécuté en avant-plan | <input type="checkbox"/> | ps grep bash sera exécuté en arrière-plan | <input type="checkbox"/> |
| grep bash sera exécuté en avant-plan | <input type="checkbox"/> | xeyes passera dans l'état <i>Stopped</i> | <input type="checkbox"/> |
| ps sera exécuté en avant-plan | <input type="checkbox"/> | firefox sera exécuté en arrière-plan | <input type="checkbox"/> |
| xeyes passera en avant-plan | <input type="checkbox"/> | ps sera exécuté en arrière-plan | <input type="checkbox"/> |

- * Cochez les affirmations vraies : le super utilisateur peut,

- | | | | |
|--|--------------------------|---|--------------------------|
| lire le contenu de tous les fichiers utilisateurs | <input type="checkbox"/> | changer les droits des fichiers des utilisateurs | <input type="checkbox"/> |
| changer le mot de passe des utilisateurs | <input type="checkbox"/> | changer les liens de parenté entre processus | <input type="checkbox"/> |
| traverser tous les répertoires | <input type="checkbox"/> | retrouver un fichier utilisateur effacé par mégarde | <input type="checkbox"/> |
| connaître le mot de passe des utilisateurs | <input type="checkbox"/> | changer un compte utilisateur en super utilisateur | <input type="checkbox"/> |

- * Expliquer ce qui différencie un processus en avant-plan d'un processus en arrière-plan

- * Quel est l'ordre de création des processus lorsque l'on tape une commande du type :

```
$ processus-1 | processus-2 | processus-3
```

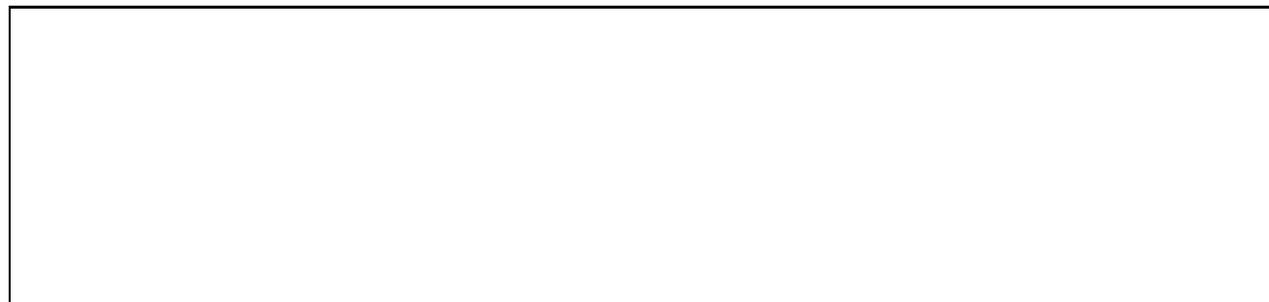
- | | | | |
|--|--------------------------|---|--------------------------|
| 1, 2 et 3 sont créés en même temps | <input type="checkbox"/> | 3 puis 2 puis 1 | <input type="checkbox"/> |
| 1 puis 2 puis 3 | <input type="checkbox"/> | Cela peut être n'importe quel ordre | <input type="checkbox"/> |

- * En supposant que martin est votre login et que vous êtes dans l'un de vos sous-répertoires :

```
~/tmp$ pwd
/home/martin/tmp
~/tmp$ ls -al
drwxr-xr-x 3 martin cremi 4096 jan 23 17:06 .
drwxr-xr-x 3 martin cremi 4096 jan 23 17:01 ..
-rw-r--r-- 1 martin cremi 17 jan 23 17:06 bar
-rw-r--r-- 1 martin cremi 17 jan 23 17:06 foo
drwxr-xr-x 2 martin cremi 4096 jan 23 17:06 XIII
~/tmp$
```

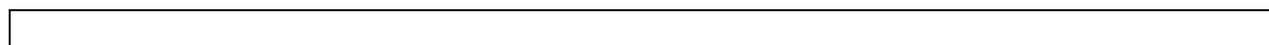
Écrire les commandes qui permettent de modifier les permissions en :

```
~/tmp$ pwd
/home/martin/tmp
~/tmp$ ls -al
drwxr---x 3 martin cremi 4096 jan 23 17:06 .
drwxr-xr-x 3 martin cremi 4096 jan 23 17:01 ..
-rwxr-xr-- 1 martin cremi 17 jan 23 17:06 bar
-rwxr-xr-- 1 martin cremi 17 jan 23 17:06 foo
drwxr-x--- 2 martin cremi 4096 jan 23 17:06 XIII
~/tmp$
```



* Que produit l'exécution des commandes suivantes

```
~$ XIII=13
~$ export TREIZE=\$XIII
~$ sh
~$ echo "$XIII*$TREIZE"
```



```
~$ exit
~$ eval echo $TREIZE
```



PROBLÈME

Le fichier HSU09584.txt contient la séquence ARNm (acide ribonucléique messager) correspondant à protéine PL6 de l'Homo Sapiens (être humain).

```
$ cat HSU09584.txt
1 ggcgaggggc ctacgctgcg gcccggaac aaggcccagc tcggcccctc gggaccagag
61 cccaccoga tcggaagcgg atcctttacc agggccatag gccagtgact aggccgggcc
121 tggacctocc atcggggccg gactaggacg agggcccggg gaggcccctg gcctaccaga
181 ccctttctc aggcgacag ccgccaggaa gatgcaacgt gccctgccag gcgcccgcca
241 gcaactgggg gccattctgg ccagcgccag cgtgggtggtg aaggctctgt gtgcggcggt
301 actattctc tacctgctct ccttcgccgt ggacacaggc tgccctggcg tcaccccggg
361 ctacctctt cctcccaact tctggatctg gaccctggcc acccatgggc tgatggagca
421 gcatgtgtgg gacgtggcca tcagcctgac aacggtggtg gtggccgggc gttgctgga
481 gccctctgg gggccttgg agctgctcat cttcttctca gtggtgaatg tgtctgtagg
541 gctgctgggg gccttcgctt acctcctcac ctacatggct tccttcaacc tggctacct
601 gttcactgtc cgtatccacg gcgccttggg cttctaggt ggcgtcctgg tggcactcaa
661 gcaaaccatg ggggactgtg tggctctgcg agtgccccag gtgcgcgtca gtgtgatgcc
721 catgctgctg ctggcgctgc tgctcctgct gcggctcgcc aactgctcc agagcccggc
781 gctggcttcc tatggcttcg gctgctctc cagtgggta tatcttgcct tctaccagcg
841 ccatagccga gcccgagggg acatggctga ccactttgct ttogccactt tcttcctga
901 gatcctgcag cctgtggtgg gtttgctggc gaacttggtg cacagcctcc tggatgaagg
961 aaagatatgc cagaagacgg tgaagcgcta cgatgtgggt gccccatcct ccatccatc
1021 cagcctgcca ggcacagacc ctcaagacgc cgagcggaga aggcaactgg ccctgaaggc
1081 actcaatgag cggctgaaga gagtgaaga ccagtccatc tggcccagca tggatgatga
1141 tgaagaggag tctggggcca agtggtgacag cccctgccc tcagacaaaag ctcccacacc
1201 cccaggaag gggcctgccc cagaatccag tctaatcacc ttogagggcag ctccccgac
1261 gctgtaactc cagaccacct tgagtgtggc acctcccctc ccaagcccc cgttgacatc
1321 ctctcagcta ctccagggca cctgactgct ctgaggagag ggaagaaggc ctgctggggc
```

```

1381 tttccatggc ottctgctgt ttctcgccaa cactaccag gactcttgct acctggttcc
1441 aactccagac aaccactatg ccaggcccgg agcctctgag gcatcggcca gtccaggccc
1501 tcatctgagg taagaatgta catcagctgg cagcccgaag caagtggctg cagggacact
1561 gatgccacag ctctctggggc ggcctcaca tctgaaactg gttgcccaga gccctgagcc
1621 aaggcaagga tttgccaaaa atgttctggg gcccagcaa atgcaggagc cgacctgggg
1681 ctgcacatcc ctgcccattc ccagaaagac tgttctctgc aggatttgtt tccctctgct
1741 gtggcggtga ctgcttctgg accagaacag ctccagctcc caggatattt ctacaggacc
1801 acttgagtgg gcagccaagc ccaggctcgc agtatcaata aagcagtctc ctgaggaatg

```

\$

Le fichier XM_001099934.txt contient la séquence ARNm correspondant à protéine PL6 du singe Macaca mulatta (rhesus monkey).

\$ cat XM_001099934.txt

```

1 ttggcctcca gcccctgt ttgtcggccc cccgagaaag aggtacggtc gaagccagga
61 gccaggccca gcgggagctg accaggcttg actcgggtac ggaactaggc atcagcccc
121 ttgcaaccg aaggcctcg gagtggatgg aggaggccca gccctgagat caacgccagc
181 caggctagcc tggcacgggg cctacagggt gggcaggcgg gcgtgccga gccgtccagg
241 gccttccctc aggtcccggg cgagggacct aactgcggc ccggcgaca ggcccgactc
301 gcccctcgg gaccagaacc ccaccgatc ggaagcggat cctttaccag ggccataggc
361 cagtgactag gccgggctg ggcctccat cggggccgga ctgggacgag gcccccggga
421 gcccctggc ctaccagacc ctttctcag gtctacagcc gccgggaaga tgcaacgtgc
481 cctgccagcc gcccgcagc acttgggggc catcctggcc agcgcagcg tgggtggtgaa
541 ggctctgtg gcagcggta tttctctta cctgctctcc ttgcgctgg acacaggctg
601 cctggcgctg accccgggtc acctcttcc cccaacttc tggatctgga cctggccac
661 ccatgggctg atggagcagc atgtgtggga cgtggccatc agcctgacca cgggtgggtg
721 ggctggcgt ttgctggagc cctctgggg ggccttgag ctgctcatct tcttctcagt
781 ggtgaatgtg tctgtagggc tgtgggggc cttcgcctac ctctcacct acatggcttc
841 cttcaacctg gtctacttgt tcaactgtcg tatccacggc gccttgggct tcttaggtgg
901 tgtcctggg gcactcaagc aaaccatggg ggactgtgtg gtctcgcgag tgcccagggt
961 gcgcgtcagt gtgatgcca tgtgctgct ggcactgctg ctctcgtgc ggctcggccac
1021 actgctccag agcccagcac tggcttcta tggcttcgga ttgctctcca gttgggtgta
1081 tttctgcttc taccagcgc acagccgggg ccgaggggac atggctgacc attttgcttt
1141 cgccactttc ttccctgaga tctgcagcc tgtggtgggg ttgctggcga acttgggtgca
1201 cggcctctg gtgaaggtaa agatatgcca gaagcggtg aagcgctacg atgtgggtgc
1261 cccatctcc atcacatca gctgcccag cacagacct caagcgcgg agcggagaag
1321 gcaactggcc ctgaaggcac tcaatgagcg gctgaagaga gtggaggacc atccgctg
1381 gcccagcagc gatgatgatg aagaggatc tggggcgaag gtgagcagcc cctgcccctc
1441 agacaaagct cctacacccc cagggaaggg ggcctgcccc gaatccagtc taatcacctt
1501 cgaggcagct ccctcgacgc tgtaactcca gaccacctg agtgtggcac ctcccctct
1561 gagccccct tgacatctc tcagctactc caggcacct gactgctctg aggagagggg
1621 agaaggcctg ctggggcttt ctatggcctt ctgctgttc tcgccaacac taccaggac
1681 tcttgtacc tggttccaac tccagacaac cactatgcca ggcctggagc ctctgaggca
1741 tcggccagtc caggtcctca tctgaggtaa gaatgtacat cagctggcag cccaagcaa
1801 gtggctcagc ggacactgat gccacagctc ctgggctggc cctcacatct gacactggtt
1861 gccgagagcc ctgagccaag gcaaggattt gccaaaaatg ttctgggggc ccagcaaag
1921 caggagccga cctggggctg cacatcctg cccatccaca gaaagactgt tctgtcagg
1981 atttgttct ctctgctgtg gccgtgactg cttctggacc agaacagctt cggctcccag
2041 gtattttcta caggacctt ttgagtgggc agccaagccc aggctcgcag tatcaataaa
2101 gcagttctct gacgaatga

```

\$

Ces fichiers ne sont malheureusement pas très exploitables tel quel par les biologistes, car il contiennent des tabulations, des espaces, des numéros de lignes, des retours à la ligne. Les biologistes souhaiteraient pouvoir travailler sur des fichiers normalisés appelé HSU09584.norm et XM_001099934.norm qui ne contiennent ni espace, ni tabulation ni retour à la ligne ou tous les nucléotides sont notés en minuscule et où la base Thymine (noté 't') est remplacé par la base azotée Uracile (qui sera notée 'u').

* Écrire (dans le cadre ci-après) un fichier script-shell **normalise-ARNm.sh** qui renvoie sur la sortie standard la séquence d'ARNm normalisé et où la base Thymine est remplacée par l'Uracile. Par exemple votre script produira :

\$./normalise-ARNm.sh HSU09584.txt

```

ggcgagggggcuacgcugcgccggcggcaacaaggcccgcacucggccccucgggaccagagccccaccggaucggaagcggauccuuuaccaggggccauaggc
cagugacuagggccgggcccuggaccucccaucggggccggaacuaggaagggcccggggaggcccucggccuaccagaccuuuucucaggccgacagccgc
caggaagaugcaacgugcccugcagggcggccgagcacuugggggccaucucggccagcgcagcuggguggugaaggcucugugugcggcgguacuauu
ccucuaaccugcucuccuugccguggacacaggcugccugcgguacccccgggcuaccuuccuucccaacuucggaucuggaccucggccaccaaug

```



```

$ cat exemple.sh
#!/bin/bash

# recupere le premier parametre passé en argument
PAS=$1
# recupere le second parametre passé en argument
FIN=$2

# initialisation de variables
NBR_DEBUT=1
NBR_FIN=$PAS

# boucle tant que, l'instruction -le signifie "inférieur ou égal à"
while [ $NBR_FIN -lt $FIN ]
do

    echo $NBR_DEBUT-$NBR_FIN

    # la commande expr permet de faire des calcul sur des entiers, ici + 1
    NBR_DEBUT=$(expr $NBR_DEBUT + 1)
    NBR_FIN=$(expr $NBR_FIN + 1)

done

```

Ce script prend deux paramètres ; le premier est la taille du pas de calcul, le second le nombre maximum à ne pas dépasser. Ce script, avec un pas de 5 et un nombre maximum de 10, produit le résultat suivant :

```

$ ./exemple.sh 5 10
1-5
2-6
3-7
4-8
5-9
6-10
$

```

- * En vous inspirant du script-shell précédent, créer un script shell **extrait-toutes-les-sous-sequences.sh** qui prend en paramètre le nom d'un fichier ARNm normalisé, et un entier n correspondant à la longueur d'une sous-séquence. Votre script renverra sur la sortie standard toutes les sous-séquences de taille n composant la séquence ARNm contenu dans le fichier passé en paramètre.

Par exemple la recherche de toute les sous-séquences de taille 5 contenu dans le fichier XM_001099934.norm se fera en invoquant la commande :

```

./extrait-une-sous-sequence.sh XM_001099934.norm 5
uuggc
uggcc
ggccu
gccuc

[...]

cgaau
gaaug
aauga

$ cat extrait-toutes-les-sous-sequences.sh
#!/bin/bash

```

* Écrire la commande en ligne qui permet de sauvegarder dans un fichier liste-sous-sequences.txt toutes les sous-séquences de longueur 100 contenu dans la séquence du fichier XM_001099934.norm

* Écrire la commande qui permet de tester si la quatorzième sous-séquences de longueur 100 contenu dans le fichier liste-sous-sequences.txt est également présente dans le fichier HSU09584.norm.

FIN.

ANNEXES

Table des matières

A	chgrp	10
B	chown	10
C	cut	10
D	eval	11
E	expand	12
F	expr	12
G	grep	14
H	sort	17
I	tr	18
J	uniq	19

Annexes – Extraits de Pages du manuel

A chgrp

CHGRP(1) User Commands CHGRP(1)

NAME

chgrp - change group ownership

SYNOPSIS

```
chgrp [OPTION]... GROUP FILE...
chgrp [OPTION]... --reference=RFILE FILE...
```

DESCRIPTION

Change the group of each FILE to GROUP. With --reference, change the group of each FILE to that of RFILE.

[...]

B chown

CHOWN(1) User Commands CHOWN(1)

NAME

chown - change file owner and group

SYNOPSIS

```
chown [OPTION]... [OWNER][:[GROUP]] FILE...
chown [OPTION]... --reference=RFILE FILE...
```

DESCRIPTION

This manual page documents the GNU version of chown. chown changes the user and/or group ownership of each given file. If only an owner (a user name or numeric user ID) is given, that user is made the owner of each given file, and the files~ group is not changed. If the owner is followed by a colon and a group name (or numeric group ID), with no spaces between them, the group ownership of the files is changed as well. If a colon but no group name follows the user name, that user is made the owner of the files and the group of the files is changed to that user~s login group. If the colon and group are given, but the owner is omitted, only the group of the files is changed; in this case, chown performs the same function as chgrp. If only a colon is given, or if the entire operand is empty, neither the owner nor the group is changed.

[...]

C cut

CUT(1) Manuel de l'utilisateur Linux CUT(1)

NOM

cut - Supprimer une partie de chaque ligne d'un fichier.

```
SYNOPSIS cut {-b liste_octets, --bytes=liste_octets} [-n]
[--help] [--version] [fichier...]
```

```
cut {-c liste_caractères, --characters=liste_caractères}
[--help] [--version] [fichier...]
```

```
cut {-f liste_champs, --fields=liste_champs} [-d séparateur]
[-s] [--delimiter=séparateur] [--only-delimited] [--help]
```

[--ver- sion] [fichier...]

DESCRIPTION

Cette page de manuel documente la version GNU de cut.

cut affiche une partie de chaque ligne de chacun des fichiers mentionnés, ou de l'entrée standard si aucun fichier n'est indiqué. Un nom de fichier '-' correspond également à l'entrée standard.

La partie affichée est sélectionnée par les options présentées ci-dessous.

OPTIONS

Les liste_d_octets, liste_de_caractères, et liste_de_champs se présentent sous forme d'un ou plusieurs nombres ou intervalles (deux nombres séparés par un tiret), séparés par des virgules. La numérotation des octets, caractères ou champs commence à 1. Des intervalles incomplets peuvent être indiqués : '-m' signifiant '1-m'; 'n-' signifiant de 'n' à la fin de la ligne.

-b, --bytes liste_d_octets

Afficher uniquement les octets aux positions indiquées dans la liste_d_octets. Les tabulations et les caractères BackSpaces sont traités comme tous les autres caractères, ils occupent 1 octet.

-c, --characters liste_de_caractères

Afficher uniquement les caractères aux positions indiquées dans la liste_de_caractères. Pour l'instant c'est équivalent à -b, mais cette option différera avec l'internationalisation. Les tabulations et les caractères BackSpaces sont traités comme tous les autres caractères, ils occupent 1 caractère.

-f, --fields liste_de_champs

N'afficher que les champs indiqués dans la liste_de_champs. Les champs sont séparés, par défaut, par une Tabulation.

-d, --delimiter séparateur

Avec -f, les champs sont délimités par le premier caractère du séparateur à la place de la Tabulation.

-n Ne pas couper les caractères multi-octets (sans effet pour le moment).

-s, --only-delimited

Avec -f, ne pas afficher les lignes qui ne contiennent pas le caractère séparateur de champs.

D eval

[...]

eval [arg ...]

The args are read and concatenated together into a single command. This command is then read and executed by the shell, and its exit status is returned as the value of eval. If there are no args, or only null arguments, eval returns 0.

[...]

E expand

EXPAND(1) User Commands EXPAND(1)

NAME

expand - convert tabs to spaces

SYNOPSIS

expand [OPTION]... [FILE]...

DESCRIPTION

Convert tabs in each FILE to spaces, writing to standard output. With no FILE, or when FILE is -, read standard input.

Mandatory arguments to long options are mandatory for short options too.

-i, --initial
do not convert tabs after non blanks

-t, --tabs=NUMBER
have tabs NUMBER characters apart, not 8

-t, --tabs=LIST
use comma separated list of explicit tab positions

--help display this help and exit

--version
output version information and exit

[...]

F expr

EXPR(1) User Commands EXPR(1)

NAME

expr - evaluate expressions

SYNOPSIS

expr EXPRESSION
expr OPTION

DESCRIPTION

--help display this help and exit

--version
output version information and exit

Print the value of EXPRESSION to standard output. A blank line below separates increasing precedence groups. EXPRESSION may be:

ARG1 | ARG2
ARG1 if it is neither null nor 0, otherwise ARG2

ARG1 & ARG2
ARG1 if neither argument is null or 0, otherwise 0

ARG1 < ARG2
ARG1 is less than ARG2

ARG1 <= ARG2
ARG1 is less than or equal to ARG2

ARG1 = ARG2
 ARG1 is equal to ARG2

ARG1 != ARG2
 ARG1 is unequal to ARG2

ARG1 >= ARG2
 ARG1 is greater than or equal to ARG2

ARG1 > ARG2
 ARG1 is greater than ARG2

ARG1 + ARG2
 arithmetic sum of ARG1 and ARG2

ARG1 - ARG2
 arithmetic difference of ARG1 and ARG2

ARG1 * ARG2
 arithmetic product of ARG1 and ARG2

ARG1 / ARG2
 arithmetic quotient of ARG1 divided by ARG2

STRING : REGEXP
 anchored pattern match of REGEXP in STRING

match STRING REGEXP
 same as STRING : REGEXP

substr STRING POS LENGTH
 substring of STRING, POS counted from 1

index STRING CHARS
 index in STRING where any CHARS is found, or 0

length STRING
 length of STRING

+ TOKEN
 interpret TOKEN as a string, even if it is a
 keyword like ~match~ or an operator like ~/~

(EXPRESSION)
 value of EXPRESSION

Beware that many operators need to be escaped or quoted for shells. Comparisons are arithmetic if both ARGs are numbers, else lexicographical. Pattern matches return the string matched between \(\ and\) or null; if \(\ and\) are not used, they return the number of characters matched or 0.

Exit status is 0 if EXPRESSION is neither null nor 0, 1 if EXPRESSION is null or 0, 2 if EXPRESSION is syntactically invalid, and 3 if an error occurred.

AUTHOR

Written by Mike Parker.

REPORTING BUGS

Report bugs to <bug-coreutils@gnu.org>.

[...]

G grep

GREP(1)

Manuel de l'utilisateur Linux

GREP(1)

NOM

grep, egrep, fgrep - Afficher les lignes correspondant à un motif donné.

SYNOPSIS

```
grep [ -[AB] ]num ] [ -[CEFGVBchilnsvwx] ] [ -e ] motif |  
-ffichier ] [ fichiers... ]
```

DESCRIPTION

grep recherche dans les fichiers d'entrée indiqués les lignes correspondant à un certain motif.

Si aucun fichier n'est fourni, ou si le nom '-' est mentionné, la lecture se fait depuis l'entrée standard.

Par défaut, grep affiche les lignes correspondant au motif.

Il existe trois variantes principales de grep, contrôlées par les options suivantes.

-G Interprète le motif comme une expression rationnelle simple (voir plus bas). C'est le comportement par défaut.

-E Interprète le motif comme une expression rationnelle étendue (voir plus bas).

-F Interprète le motif comme une liste de chaînes figées, séparées par des Sauts de Lignes (NewLine). La correspondance est faite avec n'importe laquelle de ces chaînes.

De plus, il existe deux variantes du programme : egrep et fgrep. Egrep est similaire (sans être identique) à grep -E, et est compatible avec les versions UNIX historiques de egrep. Fgrep est identique à grep -F.

Toutes les variantes de grep acceptent les options suivantes :

-num Les correspondances seront affichées avec num lignes supplémentaires avant et après. Néanmoins, grep n'affichera jamais une ligne plus d'une fois.

-A num Afficher num lignes supplémentaires après la ligne correspondante.

-B num Afficher num lignes supplémentaires avant la ligne correspondante.

-C est équivalent à -2.

-V Afficher le numéro de version de grep sur la sortie d'erreur standard. Ce numéro de version devra être inclus dans tous les rapports de bogues (voir plus bas).

-b Avant chaque ligne, afficher son décalage (en octet) au sein du fichier d'entrée.

-c Ne pas afficher les résultats normaux. À la place, afficher un compte des lignes correspondantes pour chaque fichier d'entrée. Avec l'option -v (voir plus bas), afficher les nombres de lignes ne correspondant pas au motif.

-e motif

Utiliser le motif indiqué. Ceci permet de protéger les motifs commençant par -.

-f fichier

Lire le motif dans le fichier indiqué.

-h Ne pas afficher le nom des fichiers dans les résultats lorsque plusieurs fichiers sont parcourus.

-i Ignorer les différences majuscules/minuscules aussi bien dans le motif que dans les fichiers d'entrée.

-L Ne pas afficher les résultats normaux. À la place, indiquer le nom des fichiers pour lesquels aucun résultat n'aurait été affiché.

-l Ne pas afficher les résultats normaux. À la place, indiquer le nom des fichiers pour lesquels des résultats auraient été affichés.

-n Ajouter à chaque ligne de sortie un préfixe contenant son numéro dans le fichier d'entrée.

-q Silence. Ne pas afficher les résultats normaux.

-s Ne pas afficher les messages d'erreurs concernant les fichiers inexistantes ou illisibles.

-v Inverser la mise en correspondance, pour sélectionner les lignes ne correspondant pas au motif.

-w Ne sélectionner que les lignes contenant une correspondance formant un mot complet. La sous-chaîne correspondante doit donc être soit au début de la ligne, soit précédée d'un caractère n'appartenant pas à un mot. De même elle doit se trouver soit à la fin de la ligne, soit être suivie par un caractère n'appartenant pas à un mot. Les caractères composant les mots sont les lettres, les chiffres et le souligné ('_'). ([NDT] Bien entendu les minuscules accentuées ne sont pas des lettres ! Elles servent donc à séparer les mots...)

-x Ne sélectionner que les correspondances qui occupent une ligne entière.

EXPRESSIONS RATIONNELLES (REGULAR EXPRESSIONS)

Une expression rationnelle est un motif qui permet de décrire un ensemble de chaînes. Les expressions rationnelles sont construites comme des opérations arithmétiques, en utilisant des opérateurs divers pour combiner des expressions plus petites.

Grep comprend deux versions différentes pour la syntaxe des expressions rationnelles : "simple" et "étendue." Dans la version GNU de grep, il n'y a pas de différence dans les fonctionnalités disponibles en utilisant l'une ou l'autre des syntaxes. Dans d'autres implémentations, les expressions rationnelles simples sont moins puissantes. La description ci-dessous correspond aux expressions étendues, les différences avec les expressions simples étant résumées ensuite.

Les briques élémentaires sont les expressions rationnelles correspondant à un simple caractère. La plupart des caractères, y compris toutes les lettres et les chiffres, sont des expressions rationnelles se correspondant à eux-mêmes. Tout méta-caractère ayant une signification spéciale doit être protégé en le faisant précéder d'un BackSlash.

Une liste de caractères, encadrée par [et] peut être mise en correspondance avec n'importe quel caractère unique appartenant à la liste. Si le premier caractère de la liste est l'accent circonflexe ^ alors la mise en correspondance se fait avec n'importe quel caractère absent de la liste. Par exemple, l'expression rationnelle [0123456789] convient pour n'importe quel chiffre unique.

Un intervalle de caractères ASCII peut être indiqué en donnant le premier et le dernier caractère séparés par un tiret.

Enfin, il existe certaines classes de caractères prédéfinies. Leurs noms sont assez explicites : [:alnum:], [:alpha:], [:cntrl:], [:digit:] (chiffres), [:graph:], [:lower:] (minuscules), [:print:] (affichables), [:punct:], [:space:], [:upper:] (majuscules), et [:xdigit:] (chiffres hexadécimaux). par exemple, [:alnum:] correspond à [0-9A-Za-z], à la

différence que le dernier dépend de l'encodage ASCII, alors que le premier est plus portable. Remarquez que les crochets dans les noms de classes font partie intégrante du nom symbolique, et qu'ils doivent donc être inclus en plus des crochets encadrant la liste. La plupart des méta-caractères perdent leurs significations spéciales au sein des listes. Pour inclure un] littéral, mettez- le en premier dans la liste. De même, pour inclure un ^ littéral, placez-le n'importe où sauf au début de la liste. Enfin, pour inclure un - placez-le en dernier.

Le point . correspond à n'importe quel caractère unique. Le symbole \w est un synonyme de [[:alnum:]] et \W un synonyme de [^[:alnum]].

L'accent circonflexe ^ et le symbole dollar \$ sont des méta-caractères correspondant respectivement à une chaîne vide au début et en fin de ligne. Les symboles \< et \> correspondent respectivement à une chaîne vide en début et en fin de mot. Le symbole \b correspond à une chaîne vide à l'extrémité d'un mot, et \B correspond à une chaîne vide ne se trouvant pas à une extrémité de mot.

Une expression rationnelle correspondant à un caractère unique peut être suivie par l'un des opérateurs de répétition suivants:

- ? L'élément précédent est facultatif et doit être mis en correspondance une fois au maximum.
- * L'élément précédent doit être mis en correspondance zéro ou plusieurs fois.
- + L'élément précédent doit être mis en correspondance au moins une fois.
- {n} L'élément précédent doit être mis en correspondance exactement n fois.
- {n,} L'élément précédent doit être mis en correspondance n fois ou plus.
- {,m} L'élément précédent est facultatif et doit être mis en correspondance m fois au plus.
- {n,m} L'élément précédent doit être mis en correspondance au moins n fois, mais au plus m fois.

Deux expressions rationnelles peuvent être juxtaposées, l'expression en résultant correspondra à toute chaîne formée par la juxtaposition de deux sous-chaînes correspondant respectivement aux deux sous-expression.

Deux expressions rationnelles peuvent être reliées par l'opérateur infixé | ; l'expression résultante correspondra à toute

Le point . correspond à n'importe quel caractère unique. Le symbole \w est un synonyme de [[:alnum:]] et \W un synonyme de [^[:alnum]].

L'accent circonflexe ^ et le symbole dollar \$ sont des méta-caractères correspondant respectivement à une chaîne vide au début et en fin de ligne. Les symboles \< et \> correspondent respectivement à une chaîne vide en début et en fin de mot. Le symbole \b correspond à une chaîne vide à l'extrémité d'un mot, et \B correspond à une chaîne vide ne se trouvant pas à une extrémité de mot.

Une expression rationnelle correspondant à un caractère unique peut être suivie par l'un des opérateurs de répétition suivants:

- ? L'élément précédent est facultatif et doit être mis en

correspondance une fois au maximum.
 * L'élément précédent doit être mis en correspondance zéro ou plusieurs fois.
 + L'élément précédent doit être mis en correspondance au moins une fois.
 {n} L'élément précédent doit être mis en correspondance exactement n fois.
 {n,} L'élément précédent doit être mis en correspondance n fois ou plus.
 {,m} L'élément précédent est facultatif et doit être mis en correspondance m fois au plus.
 {n,m} L'élément précédent doit être mis en correspondance au moins n fois, mais au plus m fois.

Deux expressions rationnelles peuvent être juxtaposées, l'expression en résultant correspondra à toute chaîne formée par la juxtaposition de deux sous-chaînes correspondant respectivement aux deux sous-expression.

Deux expressions rationnelles peuvent être reliées par l'opérateur infixe | ; l'expression résultante correspondra à toute chaîne correspondant à l'une ou l'autre des deux sous-expressions.

Les répétitions ont priorité sur les juxtapositions, qui à leur tour ont priorité sur les alternatives. Une sous-expression peut être entourée par des parenthèses pour modifier ces règles de priorité.

La référence inverse \n, ou n est un chiffre, correspond à la sous-chaîne déjà mise en correspondance avec la nième sous-expression rationnelle entre parenthèses.

Dans les expressions rationnelles simples, les méta-caractères ?, +, {, |, (, et) perdent leurs significations spéciales, il faut utiliser à la place leurs versions avec BackSlash \?, \+, \{, \|, \(, et \).

Dans egrep, le méta-caractère { perd sa signification spéciale, il faut utiliser \{ à la place.

H sort

SORT(1) Manuel de l'utilisateur Linux SORT(1)

NOM

sort - Trier les lignes d'un fichier texte.

SYNOPSIS

```
sort [-cmus] [-t séparateur] [-o fichier_de_sortie] [-T
répertoire_temporaire] [-bdfiMnr] [+POS1 [-POS2]] [-k
POS1[,POS2]] [fichier...]
```

[...]

-n Comparer suivant la valeur arithmétique d'une chaîne numérique initiale composée d'espaces éventuelles, suivies optionnellement du signe -, et de zéro ou plusieurs chiffres, éventuellement suivis d'un point décimal et de zéro ou plusieurs chiffres.

-r Inverser l'ordre de tri, afin que les lignes avec la plus grande valeur de clé apparaissent en premier.

-k POS1[,POS2] Une autre syntaxe possible pour indiquer les clés de tri. Les positions des champs et des caractères sont numérotées à partir de 1.

I tr

TR(1) FSF TR(1)

NAME
tr - translate or delete characters

SYNOPSIS
tr [OPTION]... SET1 [SET2]

DESCRIPTION
Translate, squeeze, and/or delete characters from standard input, writing to standard output.

-c, --complement
first complement SET1

-d, --delete
delete characters in SET1, do not translate

-s, --squeeze-repeats
replace sequence of characters with one

-t, --truncate-set1
first truncate SET1 to length of SET2

--help display this help and exit

--version
output version information and exit

SETS are specified as strings of characters. Most represent themselves. Interpreted sequences are:

\NNN character with octal value NNN (1 to 3 octal digits)

\\ backslash

\a audible BEL

\b backspace

\f form feed

\n new line

\r return

\t horizontal tab

\v vertical tab

CHAR1-CHAR2
all characters from CHAR1 to CHAR2 in ascending

order

[CHAR*]
in SET2, copies of CHAR until length of SET1

[CHAR*REPEAT]
REPEAT copies of CHAR, REPEAT octal if starting
with 0

[:alnum:]
all letters and digits

[:alpha:]
all letters

[:blank:]
all horizontal whitespace

[:cntrl:]
all control characters

[:digit:]
all digits

[:graph:]
all printable characters, not including space

[:lower:]
all lower case letters

[:print:]
all printable characters, including space

[:punct:]
all punctuation characters

[:space:]
all horizontal or vertical whitespace

[:upper:]
all upper case letters

[:xdigit:]
all hexadecimal digits

[=CHAR=]
all characters which are equivalent to CHAR

Translation occurs if -d is not given and both SET1 and SET2 appear. -t may be used only when translating. SET2 is extended to length of SET1 by repeating its last character as necessary. Excess characters of SET2 are ignored. Only [:lower:] and [:upper:] are guaranteed to expand in ascending order; used in SET2 while translating, they may only be used in pairs to specify case conversion. -s uses SET1 if not translating nor deleting; else squeezing uses SET2 and occurs after translation or deletion.

[...]

J uniq

UNIQ(1) FSF UNIQ(1)

NAME

uniq - remove duplicate lines from a sorted file

SYNOPSIS

`uniq [OPTION]... [INPUT [OUTPUT]]`

DESCRIPTION

Discard all but one of successive identical lines from INPUT (or standard input), writing to OUTPUT (or standard output).

`-c, --count`
prefix lines by the number of occurrences

`-d, --repeated`
only print duplicate lines

`-D, --all-repeated`
print all duplicate lines

`-f, --skip-fields=N`
avoid comparing the first N fields

`-i, --ignore-case`
ignore differences in case when comparing

`-s, --skip-chars=N`
avoid comparing the first N characters

`-u, --unique`
only print unique lines

`-w, --check-chars=N`
compare no more than N characters in lines

A field is a run of whitespace, then non-whitespace characters. Fields are skipped before chars.

[...]