ANNÉE 2005-2006 SEPTEMBRE 2006

Licence *Sciences Technologies* — UNIVERSITÉ BORDEAUX 1 Sciences Technologies

Étape : Master **UE :** INF103
Épreuve de : Utilisation des systèmes informatiques **Durée :** 1h30
Date : 5 septembre 2006
Heure : 15h00
Documents manuscrits et livres autorisés
Épreuve de Monsieur Delmas

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**

Indiquer votre identificateur d'anonymat : ____

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

Linux ..... ☐ Windows NT ..... ☐
NetBSD ..... ☐ Sun/Solaris ..... ☐
FreeBSD ..... ☐ MacOS X ..... ☐

⋆ Sur les systèmes de type Unix citez :

3 programmes clients de navigation web	3 programmes clients de lecture du courrier électronique	3 programmes clients de lecture des forums de discussion
–	–	–
–	–	–
–	–	–

⋆ 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                 firefox
[2]-  Stopped                 emacs
[3]+  Stopped                 ps | grep cp
[4]   Running                 xeyes &
$ bg
```

- firefox sera exécuté en avant-plan ..... ☐
- grep cp sera exécuté en avant-plan ..... ☐
- ps sera exécuté en avant-plan ..... ☐
- xeyes passera en avant-plan ..... ☐
- ps et grep cp seront exécutés en arrière-plan ..... ☐
- xeyes passera dans l'état *Stopped* ..... ☐
- emacs sera exécuté en arrière-plan ..... ☐
- ps sera exécuté en arrière-plan mais pas grep .... ☐

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

- obiwan kenobi ..... ☐
- root ..... ☐
- superuser ..... ☐
- admin ..... ☐

⋆ Cochez les affirmations vraies : le super utilisateur peut,

- lire le contenu de tous les fichiers utilisateurs .... ☐
- changer le mot de passe des utilisateurs ..... ☐
- changer les droits des fichiers des utilisateurs .... ☐
- traverser tous les répertoires ..... ☐
- changer les liens de parenté entre processus .... ☐
- connaître le mot de passe des utilisateurs ..... ☐
- retrouver un fichier utilisateur effacé par mégarde ☐
- changer un compte utilisateur en super utilisateur ☐

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

⋆ Un processus en avant-plan est prioritaire sur un processus en arrière plan.

- Vrai ..... ☐
- Faux ..... ☐

⋆ Expliquer ce qui différencie un programme exécutable d'un processus

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

Control-z ..... ☐ Escape ..... ☐
Control-c ..... ☐ Control-d ..... ☐

⋆ 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 ..... ☐ 3 puis 2 puis 1 ..... ☐
1 puis 2 puis 3 ..... ☐ Cela peut être n'importe quel ordre ..... ☐

⋆ Que produit l'exécution des commandes suivantes

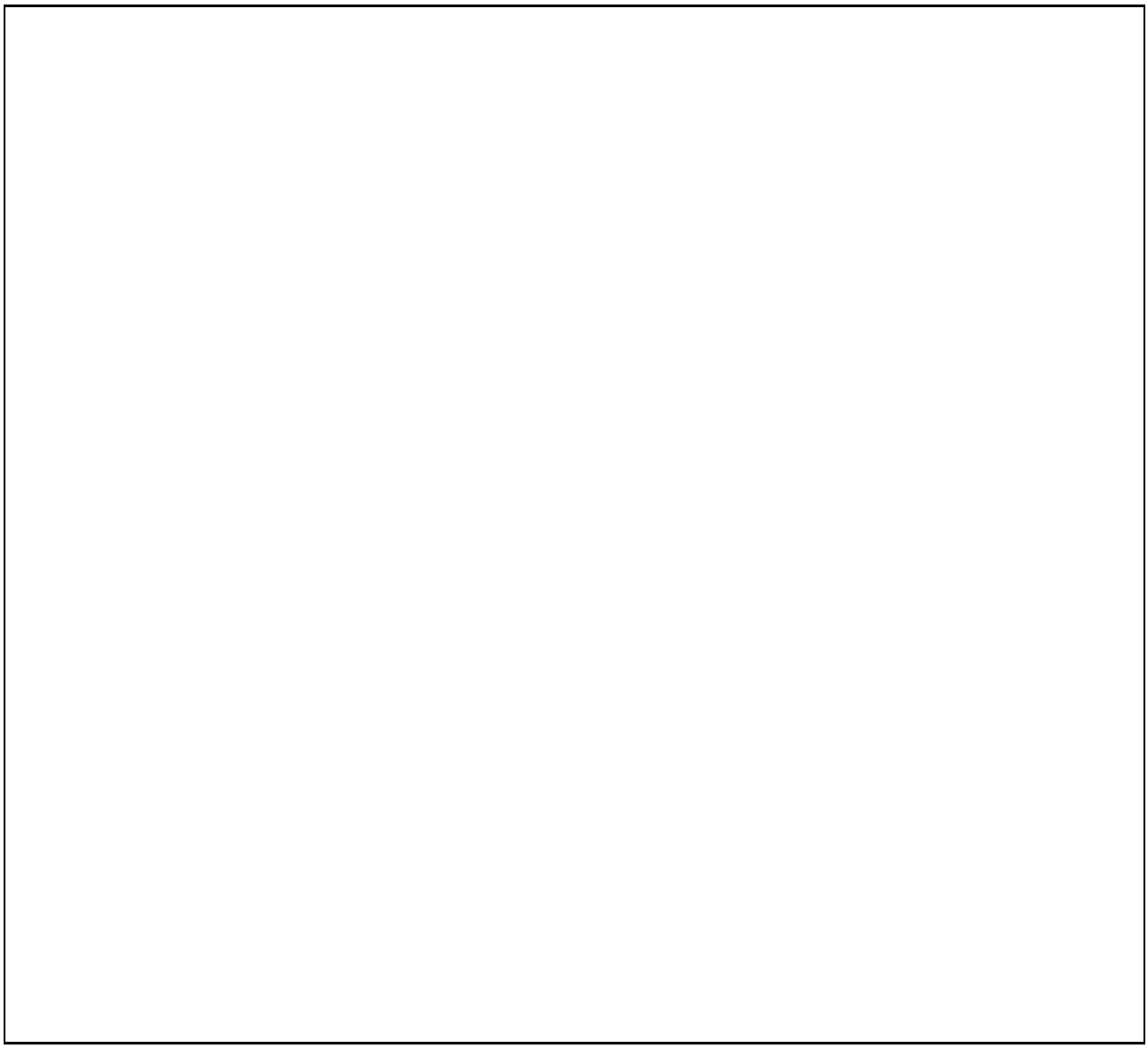
```
$ pwd
/home/martin
$ MESSAGE=vide
$ sh
$ echo "*$(pwd)$MESSAGE*"
```

```
$ exit
$ (export FOO=foo; (echo $FOO)); echo $FOO
```

```
$ echo $MESSAGE
```

⋆ Dessiner les liens de parentés à partir de votre shell ***bash*** des processus engendré par l'exécution des commandes ci-après. Vous dessinerez également dans les « patatoïdes » représentant vos processus leur table des descripteurs de fichiers (sauf pour le processus bash) et indiquerez sur quoi pointe les descripteurs 0,1 et 2.

```
$ pwd
/home/martin
$ date
mer jan 11 15:59:11 CET 2006
$ ls | more
tmp
genetique-cognitif
usi
$ (ps | head -n 1000 | more)
```



Le fichier pi.txt est un fichier qui contient les premiers milliers des décimales du nombre transcendant π. Un extrait du contenu de ce fichier est présenté ci-dessous.

```
$ head -4 pi.txt; echo "[...]"; tail -4 pi.txt
3,
14159265358979323846264338327950288419716939937 5
10582097494459230781640628620 8  998628Adfer03482534211
70679    8214808651fd ezjre32823066470938446095505822317
[...]
50201410206723   5850200724522  ADDF5632651341055924019027421 62
48439140359989535394590944070469120 9sfsd1409387001
26456001623742880210927645793106579229552 4988727584
61012648369998922569596881592056001016 55  256375678
$
```

Ce fichier débute par une ligne contenant uniquement la partie entière du nombre π c'est-à-dire `3,` puis les lignes suivantes contiennent les décimales.

⋆ Écrire les commandes permettant de créer un fichier decimales-pi.txt contenant toutes les lignes du fichier précédent sauf la première. Ainsi, on élimine du fichier la partie entière du nombre π (i.e., 3,) pour ne garder que la partie du fichier contenant les décimales de π.

Ce fichier n'est malheureusement toujours pas très exploitable tel quel par les mathématiciens, car il contient parfois de caractères alphabétiques, parfois un ou plusieurs espaces entre deux décimales, parfois des tabulations et il contient un retour à la ligne à la fin de chaque ligne. Les mathématiciens souhaiteraient pouvoir travailler sur un fichier normalisé appelé decimales-pi.dat qui ne contienne que les décimales du nombre π sans espace, ni tabulation ni retour à la ligne ni caractères alphabétiques. Voici un extrait du fichier decimales-pi.dat que l'on souhaite obtenir :

```
14159265358979323846264338327950288419716939937510582097494459230781640628620899862803482534211706798214808651328230664709384460955058223172
53594081284811174502841027019385211055596446229489549303819644288109756659334461284756482337867831652712019091456485669234603486104543266482
13393607260249141273724587006606315588174881520920962829254091715364367892590360011330530548820466521384146951941511609433057270365759591953
09218611738193261179310511854807446237996274956735188575272489122793818301194912983367336244065664308602139494639522473719070217986094370277
05392171762931767523846748184676694051320005681271452635608277857713427577896091736371787214684409012249534301465495853710507 [...]
```

Noter bien que dans ce fichier l'ensemble des décimales de π sont sur une seule et même ligne, il n'y a aucun retour à la ligne dans ce fichier.

⋆ Écrire la suite de commandes permettant d'obtenir le fichier normalisé decimales-pi.dat à partir du fichier initial decimales-pi.txt :

Les mathématiciens souhaitent faire différentes statistiques à partir de ce fichier normalisé.

⋆ Écrire la commande permettant de calculer le nombre de décimales présent dans ce fichier :

⋆ Écrire une suite de commande permettant de calculer le nombre de '0' (zéro) présent dans le fichier decimales-pi.dat :

⋆ Écrire une suite de commande permettant de calculer la proportion, i.e. le pourcentage de présence du digit '0' (zéro) dans le fichier decimales-pi.dat :

⋆ Écrire un script-shell statistiques-pi.sh affichant sur chaque ligne de la sortie standard les digits (de 0 à 9) suivi de leur nombre d'apparition dans le fichier decimales-pi.dat.

⋆ Écrire la ligne de commande appelant votre *script-shell* statistiques-pi.sh, afin que le résultat soit stocké dans un fichier decimales-pi.stat :

⋆ Écrire la commande permettant de trier les données du fichier decimales-pi.stat par ordre croissant sur le nombre d'apparition des digits.

Les mathématiciens souhaitent pouvoir travailler sur des sous-parties des décimales de π. Pour ce faire ils souhaitent travailler sur des fichiers de données ne contenant que un sous-ensembles des décimales de π. Par exemple, le fichier devra contenir les décimales de π se situant en position $11, 23$, de 112 à 1455 et celle en position 9764. **Remarque :** *Dans le fichier decimales-pi.dat la première décimale correspond à la position 1.*

⋆ Écrire un script-shell cree-sous-ensemble-decimale-pi.sh qui prend en paramètre la liste des décimales de π qui seront envoyer vers la sortie standard (cf. l'exemple situé juste après le cadre de votre réponse).

Sur l'exemple précédent votre script-shell sera appelé avec les paramètres suivants :

```
$ ./cree-sous-ensemble-decimale-pi.sh "11,23,112-1455,9764" > jeu-test-1.dat
```

On suppose que vous avez ensuite appliquer votre script-shell statistiques-pi.sh au fichier jeu-test-1.dat et que vous avez obtenu un fichier jeu-test-1.stat (et que celui-ci est trié par ordre croissant sur le nombre d'apparition des digits).

⋆ En utilisant le contenu du fichier jeu-test-1.stat, écrire une commande permettant d'afficher la liste des digits (i.e., des chiffres de 0 à 9) non présents dans le fichier (donc sans afficher leur nombre d'apparition qui bien entendu est égal à 0) :

FIN.

Annexes – Extraits de Pages du manuel

A 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 char
       acters.  Fields are skipped before chars.
[...]
```

B 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  repre
sent themselves.  Interpreted sequences are:

\NNN   character  with  octal value NNN (1 to 3 octal dig
       its)

\\     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  char
       acter   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 squeez
       ing uses SET2 and occurs after translation or deletion.
[...]
```

C 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.
```

D 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 équiva-
       lent 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.

E 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 affiches.
-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 inexistants 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 composants 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 ou 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 sym- bole \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

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 sym- bole \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.