

Répondre dans les cadres directement sur le sujet ou cocher lorsque nécessaire la ou les bonnes réponses (il peut y en avoir zéro, une ou plusieurs).

ATTENTION : cocher une mauvaise réponse retire un point.

_____ Ce sujet comporte 6 pages sans les annexes et 14 pages avec les annexes _____

Indiquer votre identificateur d'anonymat :

1 Questionnaire

Question 1.1 – Que contient traditionnellement le répertoire /usr/sbin sous Unix ?

- des bibliothèques de logiciels des programmes pour l'administrateur système ..
 des programmes pour les utilisateurs des fichiers de données

Question 1.2 – Cochez les états possibles d'un processus Unix.

- latent stoppé
 en exécution en attente
 prioritaire ininterrompible

Question 1.3 – Quelles différences y a-t-il entre un lien symbolique et un lien physique ?

- On peut créer un lien symbolique, mais non un lien physique, vers un fichier inexistant
 Un même fichier peut comporter plusieurs liens physiques, mais un seul lien symbolique peut lui être associé
 On peut créer un lien physique vers un répertoire, mais non un lien symbolique
 On peut créer un lien symbolique vers un répertoire, mais non un lien physique

Question 1.4 – En vous connectant à une machine du réseau de l'université, vous ne retrouvez pas votre répertoire d'accueil, et vous obtenez le message No directory! Logging in with home=/. Qu'a-t-il pu se passer ?

- Vous avez détruit votre répertoire d'accueil
 L'administrateur système a supprimé votre compte
 Vous vous êtes connecté par erreur comme root
 Le serveur de fichiers du réseau est en panne

Question 1.5 – Vous envisagez l'achat d'un ordinateur personnel en promotion dans une grande surface (livré avec Windows Vista pré-installé). Le chef de rayon, auquel vous expliquez que vous souhaitez pouvoir travailler sous Linux, vous tient un long discours. Parmi ses déclarations, lesquelles sont correctes ?

- La machine est livrée avec Windows Vista, mais il est possible de changer le système d'exploitation
- Cependant, le matériel étant garanti compatible avec Windows Vista, la garantie ne court pas si vous changez de système
- De toute façon, pour installer Linux sur un PC, il faut d'abord qu'il soit équipé d'un système Windows ou au minimum DOS, ne serait-ce que pour pouvoir lire le CD d'installation
- Pour installer Linux sans supprimer Windows Vista et choisir à l'amorçage (boot) quel système utiliser, vous devez configurer l'ordinateur avec un deuxième disque dur

Question 1.6 – Cochez les systèmes d'exploitation qui sont du type Unix.

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

Question 1.7 – Tous les processus se voient attribuer un PID unique par le noyau, il n'y a aucune exception :

- | | | | |
|------------|--------------------------|------------|--------------------------|
| Vrai | <input type="checkbox"/> | Faux | <input type="checkbox"/> |
|------------|--------------------------|------------|--------------------------|

Question 1.8 – Quel est le PID de l'ordonnanceur ?

- | | | | |
|---|--------------------------|-------------|--------------------------|
| -1 | <input type="checkbox"/> | 1 | <input type="checkbox"/> |
| Cela peut être n'importe quel PID | <input type="checkbox"/> | Aucun | <input type="checkbox"/> |

Question 1.9 – Sous les systèmes du types Unix, quel est le login d'usage du super utilisateur ?

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

Question 1.10 – Reliez par une flèche les cases (applications) de gauche en rapport avec celles de droite (protocoles).

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

Question 1.11 – Un processus en avant-plan est prioritaire sur un processus en arrière-plan.

- | | | | |
|------------|--------------------------|------------|--------------------------|
| Vrai | <input type="checkbox"/> | Faux | <input type="checkbox"/> |
|------------|--------------------------|------------|--------------------------|

Question 1.12 – 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"/> |

Question 1.13 – Quel est l'ordre de création des processus lorsque l'on tape une commande du type :

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

Question 1.14 – 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
```

Question 1.15 – Dessinez les liens de parenté à partir de votre shell *bash* des processus engendrés 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 pointent les descripteurs 0,1 et 2.

```
$ pwd ; (ps | head -n 10)
```

Question 1.16 – La *netiquette* est

- l'adresse d'un utilisateur à qui envoyer un courriel
- l'ensemble des règles de savoir-vivre sur l'Internet
- l'identificateur du réseau relié à votre machine ...
- l'adresse d'une page Web

Question 1.17 – Cochez les affirmations qui sont vraies.

- 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

Question 1.18 – Qu’effectuera la commande **bg** dans le contexte suivant ?

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

- | | | | |
|--|--------------------------|---|--------------------------|
| 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"/> | Cela n'aura aucun effet | <input type="checkbox"/> |

Question 1.19 – 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"/> |

2 Programmation Shell

Le fichier `e.txt` est un fichier qui contient les deux cents premiers milliards des décimales¹ du nombre transcendant e (nombre exponentiel²). Un extrait du contenu de ce fichier est présenté ci-dessous.

```
$ head -10 e.txt; echo "[...]"; tail -4 e.txt
2.
718281828459045235360287 JKd471352662497757247093699959574966967627724
07663035354759457138217852516642742746639193200305992181741359662904
3572900334295260595630738132 32862794349076323
xd38298807531952510190115
73834187930702154089149934884167509244761460668082264800168477411853
74234544243710753902!! 777449920 hehyede
6955170276183860626133138458300075204493 3826560297606737113200709328709127443747047
ss2306969772093101416928368
$
```

Ce fichier débute par une ligne contenant uniquement la partie entière du nombre e c'est-à-dire 2, puis les lignes suivantes contiennent les décimales (et des caractères parasites).

Question 2.20 – Écrivez les commandes permettant de créer un fichier **decimales-e.txt** contenant toutes les lignes du fichier précédent sauf la première. Ainsi, on élimine du fichier la partie entière du nombre e .

Ce fichier n'est malheureusement toujours pas très exploitable tel quel par les mathématiciens, car il contient parfois des caractères alphabétiques ou non-alphanumériques, 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-e.dat` qui ne contienne que les décimales du nombre e sans espace, ni tabulation ni retour à la ligne ni caractère alphabétique ou non-alphanumérique. Voici un extrait du fichier `decimales-e.dat` que l'on souhaite obtenir :

¹Record actuel du calcul des décimales de e datant du 6 mai 2009, cf. <http://ja0hvx.calico.jp/pai/estart.html>

² e est le réel tel que l'aire sous l'hyperbole $y = 1/x$ entre 1 et e vaut 1.

71828182845904523536028747135266249775724709369995957496696762772407
66303535475945713821785251664274274663919320030599218174135966290435
72900334295260595630738132328627943490763233829880753195251019011573
83418793070215408914993488416750924476146066808226480016847741185374
23454424371075390777449920695517027618386062613313845830007520449338
26560297606737113200709328709127443747047230696977209310141692836819
02551510865746377211125238978442505695369677078544996996794686445490
59879316368892300987931277361782154249992295763514822082698951936680
33182528869398496465105820939239829488793320362509443117301238197068
[...]

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

Question 2.21 – Écrire la suite de commandes permettant d'obtenir le fichier normalisé **decimales-e.dat** à partir du fichier initial **decimales-e.txt** :

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

Question 2.22 – Écrire la commande permettant de calculer le nombre de décimales présentes dans ce fichier :

Question 2.23 – Écrire une suite de commandes permettant de calculer le nombre de '0' (zéro) présents dans le fichier **decimales-e.dat** :

Question 2.24 – Écrire une suite de commandes permettant de calculer la proportion, i.e. le pourcentage de présence du digit '0' (zéro) dans le fichier **decimales-e.dat** :

Question 2.25 – Écrire un script-shell **statistiques-e.sh** affichant sur chaque ligne de la sortie standard les digits (de 0 à 9) suivis de leur nombre d'apparitions dans le fichier **decimales-e.dat**.

Question 2.26 – Écrire la ligne de commande appelant votre *script-shell* `statistiques-e.sh`, afin que le résultat soit stocké dans un fichier `decimales-e.stat` :

Question 2.27 – Écrire la commande permettant de trier les données du fichier `decimales-e.stat` par ordre croissant sur le nombre d'apparitions des digits.

Les mathématiciens souhaitent pouvoir travailler sur des sous-parties des décimales de e . Pour ce faire ils souhaitent travailler sur des fichiers de données ne contenant qu'un sous-ensemble des décimales de e . Par exemple, obtenir les décimales de e se situant en positions 11, 23, de 112 à 1455 et celle en position 9764 se fera en invoquant la commande :

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

Remarque : Dans le fichier `decimales-e.dat` la première décimale correspond à la position 1.

Question 2.28 – Écrire le script-shell `cree-sous-ensemble-decimale-e.sh` qui prend en paramètre la liste des décimales de e qui seront envoyées vers la sortie standard

On suppose que vous avez appliqué votre script-shell `statistiques-e.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'apparitions des digits).

Question 2.29 – 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'apparitions 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 characters. 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 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.

[...]

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]
[--version] [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.

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 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 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 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`, où `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.