

```

* linux/kernel/panic.c
* Copyright (C) 1991, 1992 Linus Torvalds
*/

/*
 * This function is used through-out the kernel (panic()
 * to indicate a major problem.
 */

#include <linux/config.h>
#include <linux/module.h>
#include <linux/sched.h>
#include <linux/delay.h>
#include <linux/reboot.h>
#include <linux/notifier.h>
#include <linux/init.h>
#include <linux/sysrq.h>
#include <linux/syscalls.h>
#include <linux/interrupt.h>
#include <linux/mm.h>

int panic_timeout;
int panic_on_oops;
int tainted;

EXPORT_SYMBOL(panic_timeout);

struct notifier_block *panic_notifier_list;

EXPORT_SYMBOL(panic_notifier_list);

static int __init panic_setup(char *str)
{
    panic_timeout = simple_strtoul(str, NULL, 10);
    return 0;
}

__setup("panic=", panic_setup);

/**
 * panic - halt the system
 * @fmt: The text string to print
 *
 * Display a message, then perform the functions
 * notifier list are called after the message is
 * printed. This function never returns.
 */
NMI_SAFE void panic(const char *fmt, ...)
{
    static char buf[1024];
    va_list args;
    if defined(CONFIG_BLK_DEV_S390)
        unsigned long caller = (unsigned long) __builtin_return_address(0);

    buf_start(buf);
    va_start(args, fmt);
    vsprintf(buf, size(buf), fmt, args);
    va_end(args);
    printk(KERN_EMERG "Kernel panic - not syncing: %s\n", buf);
    buf_end(buf);

    if defined(CONFIG_SMP)
        smp_send_stop();

    notifier_call_chain(panic_notifier_list, 0, buf);

    if (panic_timeout > 0)
        delay(panic_timeout);
}

```

Chapitre 7

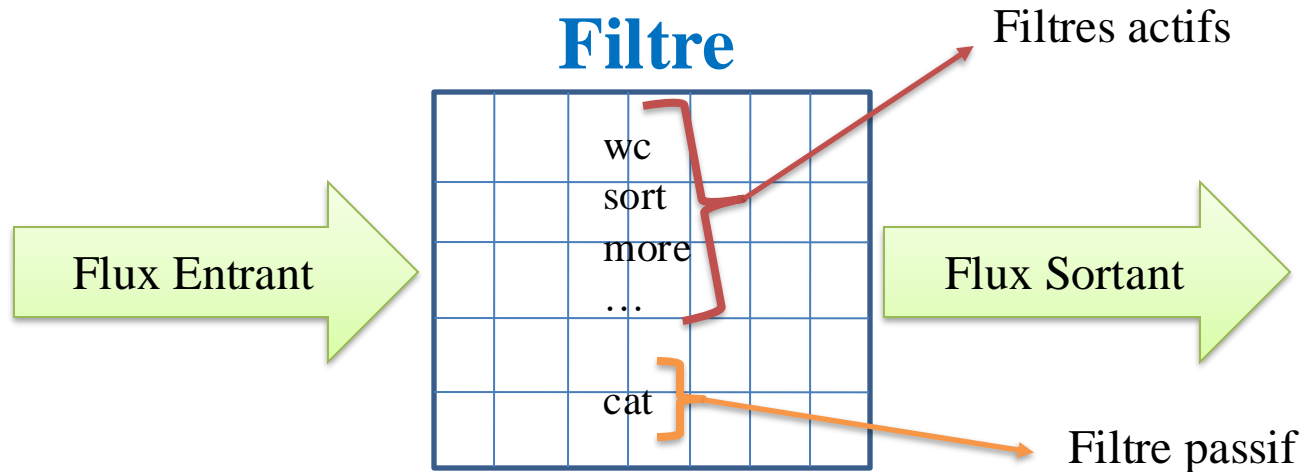
Système d'Exploitation UNIX

Les Filtres et Utilitaires

BOUKRI KHALIL

Les filtres: Introduction

Un Filtre (ou une commande filtre) est un programme sachant écrire et lire des données par les canaux standards d'entrée et de sortie. Il en modifie ou traite éventuellement le contenu.



- Les filtres sont utilisés avec les pipes (les tubes).
- Tous les processus formant un pipe tournent en même temps.

Grep

Syntaxe: *grep [Options] exp [Fichier1...]*

- ☛ Il s'agit d'extraire des lignes d'un fichier selon divers critères.
- ☛ La commande **grep** lit les données soit depuis un fichier d'entrée, soit depuis le canal d'entrée standard.

Options principales:

- v : les lignes complémentaires
- i : sans tenir compte de la casse (majuscule=minuscule)
- n : en précédant les lignes sélectionnées par leur numéro de ligne
- l : n'affiche que le nom du fichier

Expression régulières étendues

Expressions régulières par ligne:

- . : (dot) un caractère quelconque
- * : opérateur de répétition
- [x y z] : x ou y ou z
- [A-G] : intervalle
- [^xyz] : sauf x, y ou z
- ^ : début de ligne
- \$: fin de ligne
- \{m,n\} : répétitions entre m et n fois
- \ : échappement

Exemples:

- ❖ $^{\wedge}\$$: représente une ligne vide
- ❖ $^{\wedge}[A-Z]$: une majuscule en début de ligne
- ❖ $[a-z][a-z]\$$: 2 minuscules en fin de ligne
- ❖ $[ABCD]\{2,10\}\$$: entre 2 et 10 car. A,B,C ou D en fin de ligne
- ❖ $.^*$: une séquence vide ou de plusieurs caractères

Recherche de lignes dans un fichier

Exemple 1:

```
$ cat file
```

```
il fait beau
```

```
il fait chaud
```

```
beau temps n'est ce pas ?
```

```
$ grep "^il" file
```

```
il fait beau
```

```
il fait chaud
```

```
$ cat file | grep -v "^il"
```

```
beau temps n'est ce pas ?
```

```
$
```

Exemple 2:

```
$ grep "[A-Z].*[0-9]$" <<fin
```

```
> je teste la commande grep1
```

```
>La ligne 2
```

```
>Au revoir
```

```
>fin
```

☛ La ligne 2

Tr

Syntaxe: *tr [OPTIONS] chaine1 chaine2*

☛ Transposer ou éliminer des caractères substitution ou suppression de caractères sélectionnés.

Un caractère appartenant à *chaine1* est remplacé par le caractère de même position dans *chaine2*.

Options principales:

-d : suppression des caractères sélectionnés

-s : "aaaaa" dans *chaine1* → "a" dans *chaine2*

Remplacement de caractères

Format des expression de chaîne:

- **[a-z]** = segment de 26 caractères allant de 'a' à 'z'
- **[a*n]** = a. . . a (n fois)
- **\xyz** = désigne le caractère de code octal xyz
- **['[: lower:]'** = lettre minuscule
- **['[: upper:]'** = lettre majuscule

Exemples:

```
$ echo "coucou" | tr [a-z] [A-Z]
```

```
COUCOU
```

```
$ echo "BONJOUR EMSI" |tr '[:upper:]' '[:lower:]'
```

```
bonjour emsi
```

```
$ echo "aaabbbbaaa" | tr -s [a-z] [A-Z]
```

```
ABA
```


Compter les lignes: wc

Wc

Syntaxe: `wc [Options] fichier`

☛ La commande **wc** (**word count**) permet de compter les lignes, mots et caractères.

Options:

- l : compte le nombre de lignes
- c : compte le nombre d'octets
- w : compte le nombre de mots

\$ `wc liste`

12 48 234 liste

➤ Le fichier « liste » contient 12 lignes, 48 mots et 234 caractères.

Filtre Uniq

Uniq

Syntaxe: *uniq OPTIONS fichier*

☛ Élimine les lignes dupliquées dans un fichier trié par défaut.

Options principales:

- d : affiche seulement les lignes dupliquées
- u : affiche seulement les lignes non dupliquées (par défaut)
- c : donne le nombre d'exemplaires de chaque ligne

```
$ cat fichier  
un  
deux  
deux  
trois  
quatre  
quatre  
cinq  
$
```

```
$ cat fichier | uniq  
un  
deux  
trois  
quatre  
cinq  
$
```

```
$ cat fichier | uniq -d  
deux  
quatre  
$
```

Tri de lignes: sort

Sort

Syntaxe: *sort [OPTIONS] fichier*

☛ Trie les lignes d'un fichier texte

Options principales:

- n : dans l'ordre numérique
- d : dans l'ordre alphabétique
- t : le caractère séparateur de champs
- k : le champs considéré pour le tri
- o : le résultat est mis dans fichier

```
$ cat fichier
0 :Casablanca:maroc:22
1 :Rabat :maroc :37
2 :Fes :maroc :35
3 :Marrakech |:maroc :24
```

```
$ cat fichier | sort -t : -k4 -n
0 :Casablanca:maroc:22
3 :Marrakech :maroc :24
2 :Fes :maroc :35
1 :Rabat :maroc :37
```

Colonnes et champs: cut

Cut

Syntaxe: *cut [OPTIONS] fichier*

☛ Supprime une partie de chaque ligne d'un fichier par défaut

Options principales:

| | |
|------------------------------|---------------------------|
| -d séparateur : | le séparateur des champs |
| -f listes de champs : | les champs séparés |
| -s (avec -f) : | supprime les lignes vides |

```
$ cat fichier
0 :Casablanca:maroc:022
1 :Rabat :maroc :037
2 :Fes :maroc :035
3 :Marrakech :maroc :024
$
```

```
$ cat fichier | cut -d : -f1,2
0 :Casablanca
1 :Rabat
2 :Fes
3 :Marrakech
$
```

Joindre deux fichiers: join

Join

Syntaxe: `join [OPTIONS] fichier1 fichier2`

☛ Fusionner les lignes de deux fichiers ayant un champ commun.

Options principales:

-j1 n: jointure sur nième champ du premier fichier

-j2 n: jointure sur nième champ du deuxième fichier

-t car: le caractère séparateur de champs

```
$ cat file1
0 :Casablanca:maroc:022
1 :Rabat :maroc :037
2 :Fes :maroc :035
3 :Marrakech :maroc :024
$ cat file2
Touristique : 3
Industrielle : 0
Administratif : 1
$
```

```
$ join -t : -j1 1 -j2 2 file1 file2
0 :Casablanca:maroc:022: Industrielle
1 :Rabat :maroc :037: Administratif
3 :Marrakech :maroc :024: Touristique
$
```

Filtre: paste

Paste

Syntaxe: paste OPTIONS fichier...

☛ Regrouper les lignes du même n° de différents fichiers.

```
$ cat fichier1
```

```
0 :Casablanca:
```

```
1 :Rabat :
```

```
2 :Fes :maroc :35:
```

```
maroc :24:
```

```
$ cat fichier2
```

```
maroc:22
```

```
maroc :37
```

```
2 :Fes :maroc :35
```

```
3 :Marrakech
```

```
$
```

```
$ paste fichier1 fichier2
```

```
0 :Casablanca:maroc:22
```

```
1 :Rabat :maroc :37
```

```
2 :Fes :maroc :35: 2 :Fes :maroc :35
```

```
maroc :24 : 3 :Marrakech :
```

```
$
```

Autres Filtres

Head

Syntaxe: **head** OPTIONS fichier

☛ Afficher les 10 premiers lignes du fichier.

Option:

-n: Afficher les n premières lignes.

Tail

Syntaxe: **tail** OPTIONS fichier

☛ Afficher les 10 dernières lignes du fichier.

Option:

-n: Afficher les n dernières lignes.

Comparaison de fichiers

Diff

Syntaxe: `diff` [Options] fich1 fich2

- ☛ Comparer le contenu de deux fichiers.
- ☛ indique les modifications à apporter aux deux fichiers en entrée pour que leur contenu soit identique.

Option:

-b permet d'ignorer les espaces (blank)

Cmp

Syntaxe: `cmp` OPTIONS fich1 fich2

- ☛ compare les fichiers caractère par caractère. Par défaut la commande s'arrête dès la première différence rencontrée et indique la position de l'erreur.

Comparaison de fichiers

Exemple:

```
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
[emsicentreg11@localhost ~]$ cat file1
bonjour emsig1
au revoir
[emsicentreg11@localhost ~]$ cat file2
Bonjour emsig2!
Au revoir
[emsicentreg11@localhost ~]$
[emsicentreg11@localhost ~]$ diff file1 file2
1,2c1,2
< bonjour emsig1
< au revoir
---
> Bonjour emsig2!
> Au revoir
[emsicentreg11@localhost ~]$ cmp file1 file2
file1 file2 sont différents: octet 1, ligne 1
[emsicentreg11@localhost ~]$
```