

## III - Noms génériques & expansion



Voir résumé (page n° 4)

- Un caractère **spécial** est parmi :

### Caractères spéciaux

#	/	\
~	\$	
'	"	\
\$	?	!
*	?	! ^ [ ]
(.)	{.}	< > ;
&	Séparateur	

- Un **méta caractère** est de :

- **Protection** : ' ' " \

- **Substitution** : \* ? ! ^ [ ]



## nom générique

- Un **nom générique** est un mot qui contient un ou plusieurs **méta caractères** parmi : \* , ? , ! , ^ , [ , ]
  - Un **nom générique** désigne un ensemble d'objets
- Exemple de nom générique : fich\* , \*p[a-z]?
- Un nom générique est utile pour l'expansion de noms de fichiers

Application  
Exemple

Arguments d'une commande.  
ls ☐ fich\*



Voir résumé (page n° 5)

## Le méta caractère \*

Si \*

- est précédé par un caractère **séparateur** **ou**
- est précédé par le caractère "/" **ou**
- "\*" est **le premier caractère sur la ligne** de commande

alors

\* remplace n'importe quel nom de fichier y compris un nom vide (excepté un nom commençant par ".").

Sinon

\* remplace n'importe quel nom de fichier y compris un nom vide.

Fsi



Exemple de nom générique : **\*fich** ; fi\*ch



**"\*fich"** → désigne tous les noms de fichiers qui :

- **Commencent par une suite de caractères quelconque éventuellement vide (excepté ".").**
- **Se terminent par le mot "fich".**

**"fi\*ch"** → désigne tous les noms de fichiers qui :

- **Commencent par le mot "fi". Suit d'une suite de caractères quelconque éventuellement vide.**
- **Se terminent par le mot "ch".**



Voir résumé (page n° 5)

## Le méta caractère ?

**Si ?**

- est précédé par un caractère **séparateur** **ou**
- est précédé par le caractère **"/"** **ou**
- **"?"** est **le premier caractère sur la ligne** de commande

**alors**

? Remplace **un seul** caractère **non nul** dans un nom de fichier (excepté ".").

**Sinon**

? Remplace **un seul** caractère **non nul** dans un nom de fichier.

**Fsi**



Exemple de nom générique : **?a** , **fi?ch**

"?a" → désigne tous les noms de fichiers de deux caractères :

➡ **Le premier caractère est non nul et non ".".**

➡ **Le deuxième caractère est "a".**

"fi?ch" → désigne tous les noms de fichiers de 5 caractères :

➡ **Le 1° et 2° caractères sont "fi".**

➡ **Le 3° caractère est quelconque et non nul.**

➡ **Le 4° et 5° caractères sont "ch".**



Voir résumé (page n° 5)

Les méta caractères : [ ]

→ Désigne un caractère parmi un groupe de caractères.  
Ce groupe de caractères (excepté . et /) est précisé entre [ ]

Deux formes sont possibles.

→ Sous forme d'ensemble de caractères

Exemple de nom générique :

[abcd]

→ Sous forme d'intervalle de caractères

Exemple de nom générique :

[a - d]



☞ [abcd] ou [a - d] désigne tous les noms de un caractère parmi : 'a', 'b', 'c' ou 'd'.

☞ [abcd]\* ou [a - d]\* désigne tous les noms de au moins un caractère :

. Le premier caractère est parmi : 'a', 'b', 'c' ou 'd'.

. **Le reste du nom est quelconque (éventuellement vide).**





Voir résumé (page n° 6)

**Le méta caractère** `!` Ou `^` : → `!` Ou `^` : A l'aide de `!` (placé après `[`) on peut mentionner un ou plusieurs caractères qui ne doivent pas figurer dans l'intervalle ou l'ensemble.

Exemple de nom générique :

`[! a-z]` → désigne tous les noms de fichiers de un seul caractère **sauf une lettre minuscule et le caractère point.**

`[! abc]` `[! abc]` → désigne tous les noms de fichiers de deux caractères :

- Le premier caractère n'est pas parmi : 'a', 'b', 'c' et '.'.
- Le deuxième caractère n'est pas parmi : 'a', 'b' et 'c' et peut être point.
- Si `[!` est précédé par un caractère **séparateur** ou `/` alors `[!` ne prend pas ".".



Voir résumé (pages n° 5 et 6)

Les intervalles possibles sont :

<code>[0 - 9]</code>	: Les chiffres.
<code>[A - Z]</code>	: Les lettres majuscules.
<code>[a - z]</code>	: Les lettres minuscules.
<code>[0 - 9A - Z]</code>	: Les chiffres et les lettres majuscules.
<code>[0 - 9a - z]</code>	: Les chiffres et les lettres majuscules.
<code>[A - Za - z]</code>	: Les lettres majuscules et minuscules.
<code>[0 - 9A - Za - z]</code>	: L'alphanumériques.



**Remarques :**

`[1Aa - z89]` : Est l'ensemble  $\{1, A, 8, 9\} \cup \{a, \dots, z\}$

→ Pour les utiliser, ajouter dans l'environnement la variable :

**`LC_ALL=C`**

→ L'utilisation de la notion d'intervalle n'est pas portable (voir résumé) :

→ Il est conseillé d'utiliser la notion de classe.





Voir résumé (pages n° 5 et 6)

## Comment utiliser les classes ?

**Syntaxe** : `[[: nom de la classe :]]`



Exemple : `[[:lower:][:digit:]]` ou `[a-z0-9]`  
 Exemple : `[![:lower:][:digit:]]` ou `[!a-z0-9]`  
 Exemple : `[[:lower:]][:digit:]]` ou `[a-z][0-9]`  
 Exemple : `[[:lower:]][^[:digit:]]` ou `[a-z][^0-9]`



## Banalisation d'un méta caractère :



Si le caractère "\" précède un méta caractère alors ce dernier sera considéré comme tel.

Exemple:

\\* : "\*" ne sera pas expansé.  
 \\*\* : - Le premier "\*" ne sera pas expansé,  
 - Le deuxième "\*" le sera.

Si un nom générique est entre simple quote ( ' ) ou double quotes ( " ) alors ce nom générique **ne sera pas expansé**

Pour désactiver l'expansion des noms de fichier, il faut utiliser la commande :  
 set comme ceci : `set -o noglob` ou `set -f`

Pour activer l'expansion des noms de fichier, il faut utiliser la commande :  
 set comme ceci : `set +o noglob` ou `set +f`



Questions : Que représente les intervalles suivants :



[ ab- ] ; [ !a-b12 ] ; [ !adA-Def ]  
[ !! ] ; [ \! ] ; [ a\-b\\ ]



Soit "cmd" le nom d'une commande.

Comment le shell réalise l'expansion des noms de fichiers ?



**cmd** □ **nom\_générique**

- ◆ La commande "**cmd**" ne voit pas le "**nom\_générique**".
- ◆ Le shell réalise les étapes suivantes :
  - Pour chaque méta caractère du "**nom\_générique**" :  
Le shell lui applique la définition associée.
  - Le shell crée une liste (notée L ) des noms de fichiers.
  - Si la liste "**L**" est vide alors
    - le nom\_générique est conservé.
    - "**cmd** □ **nom\_générique**" est exécutée.
  - Sinon Soit "**L**" La liste "**L**" triée suivant l'ordre ASCII.
    - Le shell fournit la liste "**L**" à la commande "**cmd**".
    - "**cmd** □ **L**" est exécutée.
  - Esi

```

graph TD
    rep[rep] --> a2((a2))
    rep --> a1((a1))
    rep --> a9((a9))
    
```

**echo a[0-7]**

♦ **echo** ne voit pas **a[0-7]**

- Le shell applique la définition de **[0-7]**
- Création de **L** → **a2 a1**
- Tri de **L** → **a1 a2**
- Exécution de : **echo a1 a2**
- Affichage de :  
**a1 a2**

**ls z[0-7]**

♦ **ls** ne voit pas **z[0-7]**

- Le shell applique la définition de **[0-7]**
- Ici **L** est **vide**
- **z[0-7]** est conservé
- Exécution de : **ls z[0-7]**
- Affichage de :  
**Erreur : z[0-7] n'existe pas**

# Exemple :

## Noms génériques



On rappelle ici le résultat de la commande : **ls fich** :

On traite seulement les fichiers ordinaires et répertoires

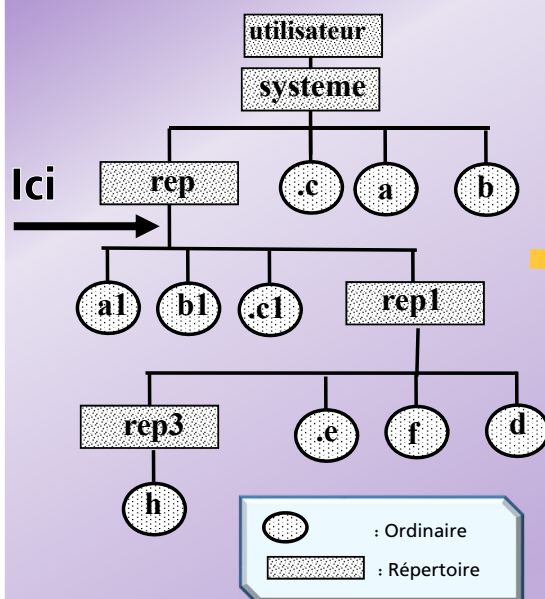
Si "fich" n'existe pas Alors "Une erreur" est affichée

Sinon Si "fich" est répertoire Alors  
le contenu visible de "fich" est affiché

Sinon "fich" est affiché

Fsi



Fsi



*Exemple* : On suppose que l'utilisateur est sous le répertoire "rep".



ls	ls _ .	ls _ . .	ls _ -a	ls _ -a _ .	ls _ -a _ . .
a1	a1	a	.	.	.
b1	b1	b	..	..	..
<u>rep1</u>	<u>rep1</u>	<u>rep</u>	.c1	.c1	.c
			a1	a1	a
			b1	b1	b
			<u>rep1</u>	<u>rep1</u>	<u>rep</u>

IUT Belfort-Montbéliard

Initiation Système Linux

M. Fouzi 19

ls _ *	ls _ .*	ls _ */*	ls _ */.*	ls _ .*/*	ls _ .*/.*
a1	.c1	rep1/d	<u>rep1/.e</u>	./a1	./c1
b1		rep1/f		./b1	../c
	./a1			../a	
rep1/d	./b1	rep1/rep3/h	rep1/./d	../b	././a1
rep1/f	./rep1		rep1/./f		././b1
Rep1/rep3			rep1/./rep3	./rep1/d	././rep1
	../a			./rep1/f	
	../b		rep1/./a1	./rep1/rep3	.././a
	../rep		rep1/./b1		.././b
			rep1/./rep1	../rep/a1	.././rep
				../rep/b1	
				../rep/rep1	.././a
					../.b
					../.rep
					.././systeme

IUT Belfort-Montbéliard

Initiation Système Linux

M. Fouzi 20

ls -a *	ls -a . *	ls -a */*	ls -a */.*	ls -a */.*	ls -a */.*	Suite de ls -a */.*
a1 b1	.c1	rep1/d rep1/f	rep1/.e	./a1 ./b1	./c1 ../c	../. ../..
rep1/.	./.	rep1/rep3/.	rep1/./.	../a	../.	.././c
rep1/..	../.	rep1/rep3/..	rep1/./..	../b	../.	.././a
rep1/.e	../c1	rep1/rep3/h	rep1/./e	./rep1/.	../.	.././b
rep1/d	./a1		rep1/./d	./rep1/..	../c1	.././rep
rep1/f	./b1		rep1/./f	./rep1/e	../a1	.././.
rep1/rep3	./rep1		rep1/./rep3	./rep1/d	../b1	.././..
	../.		rep1/./.	./rep1/f	../rep1	.././systeme
	../..		rep1/./..	./rep1/rep3	../.	
	../c		rep1/./c1	../rep/.	../.	
	../a		rep1/./a1	../rep/..	../c	
	../b		rep1/./b1	../rep/.c1	../a	
	../rep		rep1/./rep1	../rep/a1	../b	
				../rep/b1	../rep	
				../rep/rep1		

Fin cours  
Noms génériques

