

Résumé de Cours d'initiation au système Linux

I/	Table ASCII	p. 2-3
II/	Définitions préliminaires et notations	p. 4
III/	Méta caractères et substitutions	p. 5-6
IV/	La redirection	p. 7
V/	La notion de variable	p. 8

## I/ Table ASCII

[illegible]

### Table des caractères ASCII

Caractère	.	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	{	}	~		
Hexadécimal	60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F	70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F
Décimal	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
Caractère	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	_	
Hexadécimal	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F
Décimal	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
Caractère	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?	
Hexadécimal	20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F	30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F
Décimal	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
Caractère		Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
Hexadécimal	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
Décimal	Nul	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

Pour comprendre les abréviations de la première colonne, il faut lire la page suivante.

## Abréviations intervenant dans le code ASCII

NUL	— Absence de caractère.
SOH	— "Start of Heading" : début d'en-tête.
STX	— "Start of Text" : début du texte (et donc aussi fin de l'en-tête).
ETX	— "End of Text" : fin du texte, commencé avec STX.
EOT	— "End of Transmission" : fin de transmission.
ENQ	— "Enquire" : demande. Peut être utilisé pour demander au destinataire de s'identifier.
ACK	— "Acknowledge" : accusé de réception.
BEL	— "Bell" : sonnette.
BS	— "Backspace" : marche arrière de 1 caractère.
HT	— "Horizontal Tabulation" : tabulation horizontale.
LF	— "Line Fed" : retour à une nouvelle ligne.
VT	— "Vertical Tabulation" : tabulation verticale.
FF	— "Form Fed" : passage à une page suivante.
CR	— "Carriage Return" : retour du chariot.
SO	— "Shift Out" : le code suivant est hors des caractères standards. Se termine par "Shift In".
SI	— "Shift In" : retour aux caractères standard du code utilisé.
DLE	— "Data Link Escape" : change la signification d'un ou plusieurs caractères suivant. Sert à fournir des commandes supplémentaires.
DC1, DC2, DC3 et DC4	— "Device Controls..." : caractères de commande de terminaux ou périphériques.
NAK	— "Negative Acknowledgment" : réponse négative à une question.
SYN	— "Synchronous/Idle" : caractère de synchronisation ; lorsqu'aucun caractère n'est transmis, il peut être mis continûment.
ETB	— "End of Transmission Block" : fin de transmission d'un bloc de données.
CAN	— "Cancel" : annulation de la donnée précédente.
EM	— "End of Medium" : fin physique de la carte, bande, ou autre support utilisé.
SUB	— "Substitute" : remplacement.
ESC	— "Escape" : caractère de contrôle fournissant au moyen d'étendre les possibilités de codage.
FS	— "File Separator" ;
GS	— "Groupe Separator" ;
RS	— "Record Separator" ;
US	— "United Separator" : caractères de séparation à utilisation optionnelle, mais hiérarchiques, de FS (le plus général) à US.
SP	— "Space" : espace, ou blanc.
DEL	— "Delete" : suppression.

Par exemple quel est le code ASCII de retour à une nouvelle ligne ?

Réponse :

Sur la page n°2 de la table ASCII : retour à une nouvelle ligne correspond à LF

Sur la page n°1 de la table ASCII (première colonne) le code de LF est  $(10)_{10}$  ou  $(0A)_{16}$

## II/ Résumé : Définitions préliminaires et notations

<b>Caractère</b>	→	Voir table ASCII
<b>Chiffre</b>	→	0 1 2 3  ...  9
<b>Lettre majuscule</b>	→	A B C D  ...  Z
<b>Lettre minuscule</b>	→	a b c d  ...  z
<b>Caractère souligné</b>	→	—
<b>Caractère point</b>	→	.
<b>Caractère espace</b>	→	□
<b>Caractère tabulation</b>	→	└─┘
<b>Caractère retour à la ligne</b>	→	↵
<b>Caractère alphabétique</b>	→	Lettre majuscule   Lettre minuscule
<b>Caractère alphanumérique</b>	→	Lettre majuscule   Lettre minuscule   Chiffre
<b>Caractère séparateur</b>	→	Espace   Tabulation   Retour à la ligne   caractère quelconque défini par l'utilisateur

**Un mot** → Est une suite finie et non vide de caractères ne contenant pas de séparateurs.

**Un nom** → Est une suite finie et non vide de caractères alphanumérique ou "\_" ou "." commençant obligatoirement par une lettre ou "\_" ou ".".

### ⇒ Caractères spéciaux

#	→	Si# commence un mot, le reste de la ligne est un commentaire.
/	→	<b>Slash</b> : Sépare les sommets d'un chemin, le premier "/" désigne la racine d'une arborescence.
\	→	<b>Back slash</b> : Est considéré comme caractère de protection.
~	→	<b>Tilde</b> : Désigne le répertoire de connexion.
\${var}	→	Désigne la valeur de la variable <b>var</b> .
`	→	<b>Accent grave</b> : Est utile pour la substitution de commandes.
' , "	→	<b>Simple et double quotes</b> : Sont utiles pour la protection d'une expression.
[ , ] , * , ? , ! , ^	→	Sont utiles pour l'expansion du nom de fichiers.
\$ , ? , ! , * , @	→	Désigne des noms de variables spéciales.
( , ) , { , } , < , > , ;		
& , Séparateur	→	Provoquent la terminaison d'un mot.

### ⇒ Caractères de contrôles (voir la commande "stty -a")

Caractère	Résultat de stty	Description
Ctl-g	stop	Arrêt de l'affichage sur écran.
Ctl-q	start	Reprise de l'affichage sur l'écran.
Ctl-h ou del	erase	Efface le dernier caractère d'une ligne de commande.
Ctl--d	eof	Fin d'entrée de données.
Ctl-z	susp	Suspend la commande courante.
Ctl-c ou del	intr ou suppr	Arrête la commande courante
Ctl-\ ou ctl-u	quit	Arrête la commande courante plus sauvegarde l'image mémoire dans un fichier nommé "core".

**III/ Résumé : Méta caractères et substitutions :**

- Un séparateur est par défaut : espace ou tabulation ou retour à la ligne ou toutes combinaisons possibles entre ces trois caractères.

**Méta caractère : \***

\* est substitué par n'importe quelle chaîne de caractères éventuellement vide.

Si \* est le premier caractère sur une ligne de commande ou précédé par un séparateur ou précédé par / alors le nom de substitution ne doit pas commencer par le caractère point.

**Méta caractère : ?**

? est substitué par un seul caractère quelconque non nulle.

Si ? est le premier caractère sur une ligne de commande ou précédé par un séparateur ou précédé par / alors le caractère de substitution ne doit pas être le caractère point.

**Méta caractères : [ et - et ]**

[suite de caractères] : Exprime un ensemble de caractères.

[car1 - car2] : Exprime un intervalle de caractères ("car1" et "car2" sont séparés par "-").

[suite de caractères] ou [car1 - car2] est substitué par un seul caractère parmi la suite de caractère.

Si [suite de caractères] ou [car1 - car2] est le premier mot sur une ligne de commande ou précédé par un séparateur ou précédé par / alors le caractère de substitution ne prend pas le caractère point.

L'utilisation de la notion d'intervalle n'est pas portable. Il est conseillé d'utiliser la notion de classe.

Le nom d'une classe doit être indiqué par [:nom\_de\_la\_classe:]. Son utilisation se présente comme ceci : [[:nom\_de\_la\_classe:]] ou se forme d'une liste : [[:classe1:][:classe2:][:classe3:...]].

**Ci-dessous on donne les classes suivantes :**

Nom de la classe	Ascii	Commentaires
[[:upper:]]	[A-Z]	Lettres majuscules
[[:lower:]]	[a-z]	Lettres minuscules
[[:digit:]]	[0-9]	Chiffres décimaux
[[:xdigit:]]	[0-9A-Fa-f]	Chiffres hexadécimaux
[[:alpha:]]	[A-Za-z]	Alphabétique : Lettre majuscule ou minuscule
[[:alnum:]]	[A-Za-z0-9]	Alphanumérique : Lettre majuscule, minuscule ou chiffre
[[:word:]]	[A-Za-z0-9_]	Lettre majuscule ou minuscule ou chiffre ou _
[[:blank:]]	Espace ou tabulation	Blanc(Séparateur)
[[:space:]]	Espace, Tabulation, Saut de ligne, de page ou retour chariot	Séparateur
[[:punct:]]	[ ] ( ) { } & ~ " # ' -   ` _ \ ^ @ + = ^ \$ % * < > , ? . ; / : !	Caractère de ponctuation
[[:graph:]]	[[:alnum:]][[:punct:]]	Alphanumérique ou ponctuation
[[:print:]]	[[:graph:]][[:space:]]	Caractère affichable
[[:cntrl:]]	Caractères dont le codes Ascii ≤31 ou code ascii =127	Caractère non affichable
[[:ascii:]]	Les 256 caractères ascii	Voir table ascii

**Méta caractère ! Ou ^** : à placer juste après le [ :

[! suite de caractères] ou [^suite de caractères] ou [!car1-car2] ou [^car1-car2] est substitué par un seul caractère parmi le complément de la suite de caractères.

Si [! suite de caractères] ou [! car1 - car2] est le premier mot sur une ligne de commande ou précédé par un séparateur ou précédé par / alors le caractère de substitution ne prend pas le caractère point.

Remarque : Pour utiliser la notion d'intervalle classique [ ... ], ajouter dans l'environnement la variable "LC\_ALL". Il faut saisir au niveau système :

LC\_ALL=C

Exemples : Soit cmd une commande quelconque de Linux.

**Exemple n°1** : `cmd □ *fich ↵`

Ici \* est substitué par n'importe quelle chaîne de caractères éventuellement vide. Mais cette chaîne de caractères ne commence pas par le caractère point.

**Exemple n°2** : `cmd □ fi*fich ↵`

Ici \* est substitué par n'importe quelle chaîne de caractères éventuellement vide. Mais cette chaîne de caractères peut commencer par le caractère point.

**Exemple n°3** : `cmd □ ?fich ↵`

Ici ? Est substitué par un seul caractère quelconque non nulle. Mais ce caractère ne peut pas être le caractère point.

**Exemple n°4** : `cmd □ fi?ch ↵`

Ici ? Est substitué par un seul caractère quelconque non nulle. Mais ce caractère peut être le caractère point.

**Exemple n°5** : `cmd □ [[:lower:]][:digit:]] ↵` ou `cmd □ [a-z][0-9] ↵`

il s'agit de deux caractères. Le premier est minuscule et le deuxième est un chiffre.

**Exemple n°6** : `cmd □ [[:lower:]][:digit:]] ↵` ou `cmd □ [a-z0-9] ↵`

il s'agit d'un seul caractère. Celui-ci est minuscule ou chiffre.

**Exemple n°7** : `cmd □ [![:lower:]][:digit:]] ↵` ou `cmd □ [!a-z0-9] ↵`

il s'agit d'un seul caractère. Celui-ci n'est pas minuscule et n'est pas chiffre et n'est pas point.

**Exemple n°8** : `cmd □ [[:lower:]][^[:digit:]] ↵` ou `cmd □ [a-z][^0-9] ↵`

il s'agit de deux caractères. Le premier est minuscule et le deuxième est quelconque sauf un chiffre.

**Exemple n°9** : Afficher les noms de fichiers de deux caractères. Le premier est majuscule et le deuxième est minuscule.

`echo □ [[:upper:]][:lower:]] ↵` ou `echo □ [A-Z][a-z] ↵`

**Exemple n°10** : Afficher les noms de fichiers de deux caractères. Le premier est majuscule et le deuxième est minuscule ou chiffre.

`echo □ [[:upper:]][:lower:]][:digit:]] ↵` ou `echo □ [A-Z][a-z0-9] ↵`

**Exemple n°11** : Afficher les noms de fichiers de deux caractères. Le premier est majuscule et le deuxième est minuscule ou point.

`echo □ [[:upper:]][:lower:]] ↵` ou `echo □ [A-Z][a-z] ↵`

**Exemple n°12** : Afficher les noms de fichiers de deux caractères. Le premier est non majuscule et le deuxième est non minuscule.

`echo □ [^[:upper:]][^[:lower:]] □ .[^[:lower:]] ↵` ou `echo □ [^A-Z][^a-z] □ .[^a-z] ↵`



#### IV/ Résumé : la redirection :

##### Descripteurs :

- 0** → Descripteur de l'Entrée Standard e Données (**ESD**)
- 1** → Descripteur de la Sortie Standard de Résultats (**SSR**)
- 2** → Descripteur de la Sortie Standard d'Erreurs (**SSE**)

##### Redirection :

- "cmd" → est le nom d'une commande linux ou d'une commande utilisateur
- "fich" → est le nom d'un fichier ordinaire
- "n" → (l'entier  $n \geq 3$ ) : est le numéro de descripteur d'un fichier ordinaire

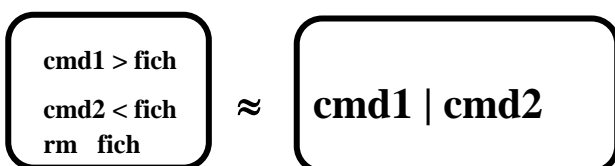
- set -C** → Active : "**noclobber**" (Si activée alors il est interdit d'écrire dans un fichier existant)
- set +C** → Désactive : "**noclobber**"

- cmd > fich** → "**fich**" est créé s'il n'existe pas, sinon son ancien contenu est détruit.  
Le résultat de "cmd" est dirigé vers "**fich**" sauf si **noclobber** est en service
- cmd >| fich** → "**fich**" est créé s'il n'existe pas, sinon son ancien contenu est détruit.  
Le résultat de "cmd" est dirigé vers "**fich**" même si **noclobber** est en service
- cmd >> fich** → "**fich**" est créé s'il n'existe pas, sinon son ancien contenu est conservé.  
Le résultat de "cmd" est ajouté dans la fin de "**fich**"
- cmd n>fich** → Redirige en **écriture** le descripteur **n** sur "**fich**" sauf si **noclobber** est en service.  
Par défaut  $n=1$
- cmd n>| fich** → Redirige en **écriture** le descripteur **n** dans le fichier "**fich**" même si **noclobber** est en service.  
Par défaut  $n=1$
- cmd n>>fich** → Redirige en **écriture** par ajout le descripteur **n** sur le fichier "**fich**". Par défaut  $n=1$ .
- cmd m>&n** → Duplique le descripteur **m** sur le descripteur **n** en écriture. Par défaut  $n=1$ .
- cmd &>fich** → Les résultats et les erreurs de "cmd" sont dirigés vers "**fich**"
- cmd < fich** → "cmd" considère "**fich**" comme entrée standard
- cmd < &n** → "cmd" considère le fichier dont le descripteur est **n** comme entrée standard
- cmd <<@** → L'entrée standard suit jusqu'à l'apparition de "@" dans le flot de données
- cmd n< fich** → Redirige en lecture le descripteur est **n** sur "**fich**" (par défaut  $n=0$ ). Si "**fich**" n'existe pas alors une erreur est affichée.
- cmd m<&n** → Duplique le descripteur **m** sur le descripteur **n** en **lecture**. Par défaut  $n=0$ . Ainsi **n** et **m** seront dirigés vers le même fichier ou périphérique.
- exec n<>fich** → Création de "**fich**" pour la **lecture** et l'**écriture** dont le numéro de descripteur est **n** ( $n \in [3-9]$ )
- exec n>fich** → Création de "**fich**" pour l'**écriture** dont le numéro de descripteur est **n** ( $n \in [3-9]$ )
- exec n<fich** → Création de "**fich**" pour la **lecture** dont le numéro de descripteur est **n** ( $n \in [3-9]$ )
- exec n>&-** → Ferme la **sortie** du descripteur **n**
- exec n<&-** → Ferme l'**entrée** du descripteur **n**

**cmd1 | cmd2** → La sortie de la commande "**cmd1**" sera l'entrée de la commande "**cmd2**" (pipeline)

**cmd1 | tee [-a] fich | cmd2** → Identique à la commande "**cmd1 | cmd2**" mais de plus, la commande **tee** mémorise les résultats de la commande cmd1 dans le fichier "fich". Pour ajouter dans le fichier "fich", la commande **tee** utilise l'option "-a"

"**cmd1**" et "**cmd2**" sont deux commandes différentes. L'exécution des trois commandes : "**cmd1>fich**", "**cmd2 < fich**" et "**rm fich**" sont presque équivalente à l'exécution de "**cmd1 | cmd2**"



## V/ La notion de variable

→ Le seul type admet en shell est le type **chaîne de caractères**. Donc **pas de déclaration de variable**. Pour utiliser les variables numériques il faut utiliser les commandes Linux (**bc** ou **expr**) ou commandes internes (**let** ou **declare**).

→ Trois Catégories de **variables** :

- **Utilisateur** son nom peut être  $\{a-z\} \cup \{0-9\} \cup \{.,\_ \}$  et ne commence pas par un chiffre.
- **Shell** son nom est en majuscule (par exemple : PWD, PATH, ...).
- **Spéciale** définit par le shell (par exemple : \$, \*, #, ...).

→ Si var est le nom de la variable alors **\${var}** est la valeur de var.

→ Les attributs d'une variable est lecture seule (voir **readonly**) et/ou exportabilité (voir **export**)

- Si une ligne commence par **#** alors le reste de la ligne est un commentaire.
- **read v1 v2 ... #** Affectation par **lecture** via l'entrée standard des variables v1, v2, etc.
- **readonly v1 v2 ... #** Les constantes v1, v2, etc sont accessibles uniquement en lecture.
- **export v1 v2 ... #** Une **copie** de v1, v2, etc sont exportées du **processus père vers le fils**.

→ Affectation

```
w=pwd      # Pas de protection : w ← pwd (v est affectée par le mot "pwd")
v=${w}     # Pas de protection : v ← pwd (v est affectée par la valeur de w)
v="$w"     # Utilisation de double quote : v ← pwd (v est affectée par la valeur de w)
v='{w}'    # Utilisation de simple quote : v ← ${w} (v est affectée par la valeur de $w)
v='pwd'    # Utilisation de accent grave : v ← le résultat de la commande pwd
v=$(pwd)   # v ← le résultat de la commande pwd
v=${w}     # w est substituée par "pwd" et v ← le résultat de la commande pwd
```

→ Tableau de protection :

Est interprété ?	' : Simple quote	" : Double quote	` : Accent grave	\$	Meta-car.	\ Anti-slash
' ... '	Oui	Non	Non	Non	Non	Non
" ... "	Non	Oui	Oui	Oui	Non	Oui
` ... ` ou \$(...)	Oui	Oui	Oui	Oui	Oui	Oui
Pas de protection	Oui	Oui	Oui	Oui	Oui	Oui