

# Unix / Linux - Substitution Shell

## Qu'est-ce que la substitution?

Le shell effectue une substitution lorsqu'il rencontre une expression qui contient un ou plusieurs caractères spéciaux.

### Exemple

Ici, la valeur d'impression de la variable est remplacée par sa valeur. En même temps, "**\n**" est remplacé par une nouvelle ligne –

[Démonstration en direct](#)

```
#!/bin/sh
```

```
a=10
```

```
echo -e "Value of a is $a \n"
```

Vous recevrez le résultat suivant. Ici le **-e** l'option permet l'interprétation des échappements de barre oblique inverse.

```
Value of a is 10
```

Suivre est le résultat sans **-e** option –

```
Value of a is 10\n
```

Les séquences d'échappement suivantes qui peuvent être utilisées dans la commande echo –

Sr.No.	Échapper et description
1	<b>\</b> barre oblique inverse
2	<b>\a</b> alerte ( BEL )
3	<b>\b</b> arrière-plan
4	<b>\c</b> supprimer la nouvelle ligne de fin
5	<b>\f</b> forme d'aliment
6	<b>\n</b> nouvelle ligne
7	<b>\r</b> retour chariot
8	<b>\t</b> onglet horizontal
9	<b>\v</b> tabulation verticale

Vous pouvez utiliser le **-E** option pour désactiver l'interprétation des échappements de barre oblique inverse ( par défaut ).

Vous pouvez utiliser le **-n** option pour désactiver l'insertion d'une nouvelle ligne.

## Substitution de commande

La substitution de commande est le mécanisme par lequel le shell effectue un ensemble de commandes donné, puis remplace sa sortie à la place des commandes.

### Syntaxe

La substitution de commande est effectuée lorsqu'une commande est donnée comme –

```
`command`
```

Lorsque vous effectuez la substitution de commande, assurez-vous d'utiliser le guillemet arrière, pas le caractère de citation unique.

### Exemple

La substitution de commande est généralement utilisée pour attribuer la sortie d'une commande à une variable. Chacun des exemples suivants démontre la substitution de commande –

[Démonstration en direct](#)

```
#!/bin/sh
```

```
DATE=`date`  
echo "Date is $DATE"
```

```
USERS=`who | wc -l`  
echo "Logged in user are $USERS"
```

```
UP=`date ; uptime`  
echo "Uptime is $UP"
```

Lors de l'exécution, vous recevrez le résultat suivant –

```
Date is Thu Jul  2 03:59:57 MST 2009  
Logged in user are 1  
Uptime is Thu Jul  2 03:59:57 MST 2009  
03:59:57 up 20 days, 14:03,  1 user,  load avg: 0.13, 0.07, 0.15
```

## Substitution variable

La substitution de variables permet au programmeur shell de manipuler la valeur d'une variable en fonction de son état.

Voici le tableau suivant pour toutes les substitutions possibles –

Sr.No.	Forme et description
1	<b><code>\${ var }</code></b> Remplacer la valeur de <i>var</i> .
2	<b><code>\${ var: -word }</code></b> Si <i>var</i> est nul ou non défini, <i>mot</i> est remplacé par <b>var</b> . La valeur de <i>var</i> ne change pas.
3	<b><code>\${ var: = mot }</code></b> Si <i>var</i> est nul ou non défini, <i>var</i> est défini sur la valeur de <b>mot</b> .
4	<b><code>\${ var:?message }</code></b> Si <i>var</i> est nul ou non défini, <i>message</i> est imprimé en erreur standard. Cela vérifie que les variables sont correctement définies.
5	<b><code>\${ var: + mot }</code></b> Si <i>var</i> est défini, <i>mot</i> est substitué à <i>var</i> . La valeur de <i>var</i> ne change pas.

## Exemple

Voici l'exemple pour montrer divers états de la substitution ci-dessus –

[Démonstration en direct](#)

```
#!/bin/sh
```

```
echo ${var:-"Variable is not set"}  
echo "1 - Value of var is ${var}"
```

```
echo ${var:= "Variable is not set"}  
echo "2 - Value of var is ${var}"
```

```
unset var  
echo ${var:+ "This is default value"}  
echo "3 - Value of var is $var"
```

```
var="Prefix"  
echo ${var:+ "This is default value"}  
echo "4 - Value of var is $var"
```

```
echo ${var:? "Print this message"}  
echo "5 - Value of var is ${var}"
```

Lors de l'exécution, vous recevrez le résultat suivant –

```
Variable is not set  
1 - Value of var is  
Variable is not set  
2 - Value of var is Variable is not set  
  
3 - Value of var is  
This is default value  
4 - Value of var is Prefix  
Prefix  
5 - Value of var is Prefix
```