

Présentation fonction AWK

The **AWK**
Programming
Language 

Lucy Edouard
Wissocq Geoffroy



SOMMAIRE

I. Les différents types de fonctions

- A) C'est quoi AWK ?
- B) Les fonctions mathématique
- C) Les fonctions personnelles
- D) Les fonctions de chaîne de caractère
- E) Conclusion

A. C'est quoi AWK ?

AWK :

- Langage de traitement de lignes
- Utilisation de manipulation de fichiers
- Opérations de recherches
- Transformations complexes

B. Les fonctions mathématique

Quelques fonctions mathématique:

-**atan2(y,x)**: arctangente de y/x en radian

-**cos(x)**: cosinus de l'angle x (radian)

-**sin(x)**: sinus de l'angle x (radian)

-**exp(x)**: exponentielle à la puissance x

-**int(x)**: garde la partie entière de x

-**sqrt(x)**: racine carrée de x

Exemple avec la fonction int:

```
root@kali: ~  
Fichier  Édition  Affichage  Rechercher  Terminal  Aide  
root@kali:~# echo 1123.456789 | awk '{print int($1)}'  
1123  
root@kali:~#
```

Exemple avec la fonction sqrt:

```
root@kali: ~  
Fichier  Édition  Affichage  Rechercher  Terminal  Aide  
root@kali:~# echo 9 | awk '{print sqrt($1)}'  
3  
root@kali:~#
```

C. Les fonctions personnelles

La fonction personnelle:

- mot-clé *function* suivi du nom de la fonction et de ses paramètres.
- mot-clé *function* suivi du nom de celle-ci "magauss" et de ses paramètres (x, a)

```
{
    x=$1
    a=$2
    y=magauss(x,a)
    print "magauss(",x,";",a,") =",y
}
function magauss(x,a){
    pi=3.1415927
    return (1/(pi*a^2))^(1/4) * exp(x) * exp(-x^2/(2*a^2))
}
```

```
$ echo "12 3" | awk -f test.awk
magauss( 12 ; 3 ) = 23.6772
```

D. Les fonctions de chaînes de caractères

Quelques fonctions de chaînes de caractères:

-length(str): Retourne la longueur du caractère *str*. Si *str* n'est pas indiquée, par défaut \$0 est utilisé.

```
root@kali: ~/Documents
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
root@kali:~/Documents# cat test
test pour la présentation awk
root@kali:~/Documents#
root@kali:~/Documents# awk '{print length($0)}' test
30
root@kali:~/Documents#
```

-index(str, st): Retourne la position du caractère *st* dans la chaîne *str*, ou 0 si non trouvé.

```
root@kali: ~/Documents
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
root@kali:~/Documents# cat test
test pour la présentation awk
root@kali:~/Documents# awk '{print index($0,"p")}' test
6
root@kali:~/Documents#
```

Ici "p" à la position 6

D. Les fonctions de chaînes de caractères

-tolower(str): Met en minuscules toute la chaîne *str* et retourne la nouvelle chaîne.

```
root@kali: ~
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
root@kali:~# echo MIN | awk '{print tolower($1)}'
min
root@kali:~#
```

-toupper(str): Met en majuscule toute la chaîne *str* et retourne la nouvelle chaîne.

```
root@kali: ~
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
root@kali:~# echo min | awk '{print toupper($1)}'
MIN
root@kali:~#
```

D. Les fonctions de chaînes de caractères

Exemple:

Fonction **gsub()**: Substitue chaque chaîne de caractère par une autre.

```
root@kali:~/Documents# cat test
L1 test pour la présentation awk
L2 test pour la commande gsub
L3 Test pour la présentation awk

root@kali:~/Documents# awk 'gsub(/présentation/, "commande") {print $0}' test
L1 test pour la commande awk
L3 Test pour la commande awk
root@kali:~/Documents#
```

E. Conclusion

AWK offre tous les outils d'un langage évolué :

- Fonctions
- Tableaux
- Des structures de contrôle comme while, if, for.
- Opérateurs arithmétiques.

Il est très bien adapté à la transformation et recherche dans les fichiers
AWK possède des fonctions prédéfinies comme print par exemple.

Les inconvénients :

Moins puissant que ses concurrents tels que Perl.
Son langage proche du C peut devenir déroutant pour les non initiés