

**Interpréteur de commandes****TP n° : 00****1) Définition**

Un interpréteur de commandes est un logiciel système faisant partie des composants de base d'un système d'exploitation. Sa fonction est d'interpréter les commandes qu'un utilisateur tape au clavier dans l'interface en ligne de commande.

**2) Historique**

Les premiers systèmes capables d'interpréter des lignes de commandes sont apparus au début des années 1960, en même temps que le clavier informatique.

**3) Quelques interpréteurs populaires****a) Interpréteur sous Windows**

- Le programme COMMAND.COM sous MS-DOS, qui peut paraître désuet, mais qui occupe un minimum de place sur disque et en mémoire.
- Le programme cmd.exe sous Windows NT et ses dérivés.
- Windows PowerShell est le nouvel interpréteur de Microsoft, avec l'approche objet de .NET 2.0.
- L'Explorateur Windows (Explorer.exe) est l'interpréteur utilisé par l'interface graphique actuelle de Windows.
- Progman.exe était l'interpréteur graphique de Windows 3.x.

En fait, dans la base de registre de Windows, de très nombreux programmes sont considérés comme des interpréteurs.

**b) Interpréteurs Unix****Shell Unix**

Un Shell Unix est un interpréteur de commandes destiné aux systèmes d'exploitation Unix et de type Unix qui permet d'accéder aux fonctionnalités internes du système d'exploitation. Il se présente sous la forme d'une interface en ligne de commande accessible depuis la console ou un terminal (tty). L'utilisateur lance des commandes sous forme d'une entrée texte exécutée ensuite par le Shell.

Les systèmes d'exploitation de type Unix disposent le plus souvent d'un Shell. À l'origine, l'interpréteur de commandes par défaut était sh, qui donna naissance à de nombreuses variantes, dont csh, étendu en tcsh, ou ksh, ou encore rc... Mais aujourd'hui bash, s'inspirant de sh, ksh, et csh, est le Shell le plus répandu, bien qu'il existe d'autres interpréteurs de commandes, comme zsh, ou ash.

**3) Les terminaux****Terminal virtuel tty**

Un terminal virtuel, /dev/tty\* est un terminal lancé généralement par le processus init. La plupart du temps, six terminaux virtuels sont lancés

automatiquement lors de la séquence de démarrage, via le script /etc/inittab, vous y accédez par le raccourci <ctrl> + <alt> + <Fn> (Fn correspond aux touche de fonction numérotée du clavier), un autre est lancé par xorg sur F7 et si vous ouvrez plusieurs sessions utilisateur, chacune utilisera le terminal virtuel suivant.

Pour chaque terminal virtuel créé par inittab, le programme login est lancé en attente. Une fois que vous avez rentré votre nom d'utilisateur et son mot de passe, si ceux-ci sont valides, login lance l'interpréteur de commandes.

**4) Langage de script**

Un langage de script est un langage de programmation interprété qui permet de manipuler les fonctionnalités d'un système informatique configuré pour fournir à l'interpréteur de ce langage un environnement et une interface qui déterminent les possibilités de celui-ci. Le langage de script peut alors s'affranchir des contraintes des commandes de bas niveau — prises en charge par l'intermédiaire de l'interface — et bénéficier d'une syntaxe de haut niveau.

Les langages de script sont généralement exécutés à partir de fichiers (dits, précisément, scripts) contenant le code source du programme qui sera interprété. Historiquement, ils ont été créés pour raccourcir le processus traditionnel de développement édition-compilation-édition des liens-exécution propre aux langages compilés. Les premiers langages étaient souvent appelés « langage de commande » ou « langage d'enchaînement des travaux » (JCL : Job Control Language) car ils permettaient simplement d'automatiser une succession de commandes simples, à la manière d'un « script » de théâtre. Par la suite, ils furent munis d'exécutions conditionnelles implicites (IBM 1130) ou explicites (JCL), et enfin d'ordres de boucle et d'opérateurs les transformant en quasi-langages de programmation.

**Propriétés**

Dans le sens le plus traditionnel, qui est celui des scripts Shell, un script sert principalement à lancer et coordonner l'exécution de programmes.

- Dans sa version la plus simple, un script ne spécifie qu'une suite de programmes à appeler dans un ordre donné (par exemple éditeur de texte, compilateur, éditeur de liens et exécution du code objet). Cela crée rapidement des inefficiencies : à quoi bon en effet tenter de charger et d'exécuter un programme dont la compilation a échoué ?
- On y ajoute donc assez vite une possibilité d'exécution conditionnelle simple (||, &&) en

fonction du résultat de l'étape immédiatement précédente (il ne sert à rien d'exécuter si la compilation est mauvaise).

- La possibilité d'y employer des variables, des paramètres, des structures de contrôle (répétition, exécution conditionnelle), etc. fait des langages de scripts de véritables langages de programmation.

## 5) Liste des langages de script

Les langages de script les plus connus sont :

- Script Shell : sh ; bash ; ksh ; zsh ; csh ; tcsh ; fish pour GNU/Linux et Unix ;
- Cmd (anciennement Command), Windows PowerShell pour Windows ;
- JavaScript ;
- Google Apps Script;
- AppleScript (peut être compilé) ;
- VBScript ;
- Lisp ; Scheme (peuvent être compilés) ;
- GNU Octave
- Matlab
- PHP (peut être compilé) ;
- AutoIt (peut être compilé) ;
- Perl (peut être compilé) ;
- Python (peut être compilé) ;
- Rebol ;
- Rexx ;
- Ruby ;
- Lua ;
- Tcl/Tk;
- KiXtart ;
- Groovy ;
- etc.

Mais il en existe beaucoup d'autres.

## 6) Utilisation des scripts

Les scripts sont utilisés à différents niveaux :

- côté OS comme une méthode d'automatisation (VBA de Microsoft, Python) ;
- côté serveur web dits Server Side Includes comme une méthode pour générer des pages dynamiques (JSP, PHP, ASP, ASP.NET) ;
- côté navigateur dits Client Side sous forme d'Applet (actionscript, C#, Applet java) d'automatisation (VBA de Microsoft). C'est un langage binaire ;
- interface graphique dit gui scripts, permettant de simuler des déplacements de la souris et des pressions de touches sur le clavier.

## 7) Introduction au Script Bash

Bash est l'un des outils de scripting les plus populaires disponibles sous Unix. C'est l'acronyme de Bourne Again Shell. C'est un outil puissant pour tout utilisateur de Linux ou administrateur système. Apprenons à utiliser Script Bash !

Unix a 2 grandes catégories de Shell.

- Bourne-type Shell
- C Shell

Bourne Shell est également classé comme :

- Korn Shell (ksh)
- Bourne Shell (sh)
- POSIX Shell (sh)
- Bourne Again Shell (bash)

C Shell est également classé comme :

- C Shell (csh)
- TENEX (TOPS) C Shell (tcsh)

Les scripts Bash sont un élément extrêmement puissant et utile pour le développement. Il permet de raccourcir les tâches répétitives en un seul appel de fonction. De nombreuses commandes longues peuvent être consolidées en un seul code exécutable.

Bash est facilement disponible sur presque toutes les versions de Linux et ne nécessite aucune installation séparée. La liste des Shell disponibles peut être vérifiée en tapant la commande suivante :

cat /etc/shells

Le résultat sera quelque chose comme indiqué ci-dessous :

/bin/bash

/bin/sh

/bin/tcsh

/bin/csh

## 8) Pourquoi utiliser les fonctions Bash ?

Les fonctions Bash peuvent :

- Éliminer les tâches répétitives
- Gagner du temps
- Donne une séquence d'activités bien structurée, modulaire et organisée.
- Avec les fonctions, nous pouvons fournir des valeurs dynamiques aux commandes en utilisant des arguments de ligne de commande
- Peut simplifier les commandes complexes en une seule opération
- Une fois créé, il peut être exécuté un nombre illimité de fois par n'importe qui
- Les flux logiques peuvent être créés en utilisant des fonctions bash

• Les fonctions Bash peuvent être exécutées au démarrage du serveur ou en ajoutant une tâche cron programmée

• Les commandes peuvent être déboguées

• Peut avoir des commandes Shell interactives

Bash est sans aucun doute un excellent outil pour faciliter votre travail et améliorer votre projet. Les possibilités d'utilisation sont illimitées.

## OCM

### Question 1 : Quelle est la fonction principale d'un interpréteur de commandes ?

A) Jouer à des jeux vidéo

B) Gérer les pilotes de périphériques

C) Interpréter les commandes tapées par l'utilisateur dans l'interface en ligne de commande

D) Calculer des équations mathématiques

### Question 2 : Quand ont commencé à apparaître les premiers systèmes capables d'interpréter des lignes de commandes ?

A) Dans les années 1990

B) Dans les années 1980

C) Au début des années 1960

D) Dans les années 2000

### Question 3 : Quel composant d'un système d'exploitation est responsable de l'interprétation des commandes de l'utilisateur ?

- A) Le navigateur web
- B) L'éditeur de texte
- C) L'interpréteur de commandes
- D) Le gestionnaire de fichiers

**Question 4 : Quel est le Shell le plus répandu sur les systèmes d'exploitation de type Unix ?**

- A) csh
- B) ksh
- C) bash
- D) zsh

**Question 5 : Quelle est la principale différence entre un interpréteur sous Windows et un Shell Unix ?**

- A) Les interpréteurs Windows sont plus légers en mémoire.
- B) Les interpréteurs Unix sont destinés à l'interface graphique.
- C) Les interpréteurs Windows utilisent .NET 2.0.
- D) Les interpréteurs Unix permettent d'accéder aux fonctionnalités internes du système d'exploitation via une interface en ligne de commande.

**Question 6 : Combien de terminaux virtuels sont généralement lancés automatiquement lors de la séquence de démarrage d'un système?**

- A) Un
- B) Deux
- C) Six
- D) Dix

**Question 7 : Comment accède-t-on à un terminal virtuel sur un système?**

- A) En appuyant sur <ctrl> + <alt> + F7
- B) En ouvrant une session utilisateur
- C) En exécutant le programme login
- D) En utilisant le raccourci <ctrl> + <shift> + T

**Question 8 : Qu'est-ce qui se passe une fois que vous avez entré votre nom d'utilisateur et votre mot de passe dans un terminal virtuel?**

- A) Le système redémarre
- B) Le programme login est lancé en attente
- C) Vous êtes déconnecté du terminal virtuel
- D) L'interpréteur de commandes est directement lancé

**Question 9 : Quelle est la principale caractéristique des langages de script par rapport aux langages compilés?**

- A) Les langages de script nécessitent une compilation avant l'exécution
- B) Les langages de script sont plus efficaces en termes de performances
- C) Les langages de script sont interprétés et n'ont pas besoin d'une étape de compilation séparée
- D) Les langages de script sont principalement utilisés pour le développement de jeux vidéo

**Question 10 : Quelle est la principale fonction d'un script dans sa version la plus simple?**

- A) Coordonner l'exécution de programmes
- B) Compiler le code source
- C) Exécuter des boucles complexes
- D) Créer des interfaces utilisateur graphiques

**Question 11 : Dans quel contexte les scripts peuvent-ils être utilisés?**

- A) Pour jouer à des jeux vidéo
- B) Pour compiler du code source
- C) Pour générer des pages web dynamiques
- D) Pour simuler des déplacements de souris

**Question 12 : Quelle est la principale fonction des scripts côté OS?**

- A) Générer des pages web dynamiques
- B) Simuler des déplacements de souris
- C) Automatiser des tâches système
- D) Jouer à des jeux vidéo

**Question 13 : Quel est l'acronyme de Bash?**

- A) BASH
- B) Bourne Again Shell
- C) C Shell
- D) Korn Shell

**Question 14 : Quelle commande peut être utilisée pour vérifier la liste des Shells disponibles sur un système Unix?**

- A) cat /etc/shells
- B) listshells
- C) checkshells
- D) shellslist

**Question 15 : Quel est l'un des avantages mentionnés pour utiliser les fonctions Bash?**

- A) Créer des commandes complexes
- B) Gagner du temps et éliminer les tâches répétitives
- C) Rendre les scripts illisibles
- D) Nécessiter une installation séparée

**Question 16 : À quoi peuvent servir les fonctions Bash?**

- A) Créer des pages web dynamiques
- B) Gérer des bases de données
- C) Fournir des valeurs dynamiques aux commandes en utilisant des arguments de ligne de commande
- D) Jouer à des jeux vidéo

**Question 17 : Quel est le principal avantage de l'utilisation de Bash?**

- A) Il est facilement disponible sur toutes les versions de Linux
- B) Il peut être exécuté uniquement par l'administrateur système
- C) Il ne peut pas être utilisé pour créer des flux logiques
- D) Il nécessite une installation séparée sur Linux