

Systèmes UNIX. Architecture

Sergey Kirgizov

Philosophie d'Unix

Littérature

Architecture

Accès à un système UNIX

Notion de commandes

Variables d'environnement

La documentation UNIX

Douglas McIlroy



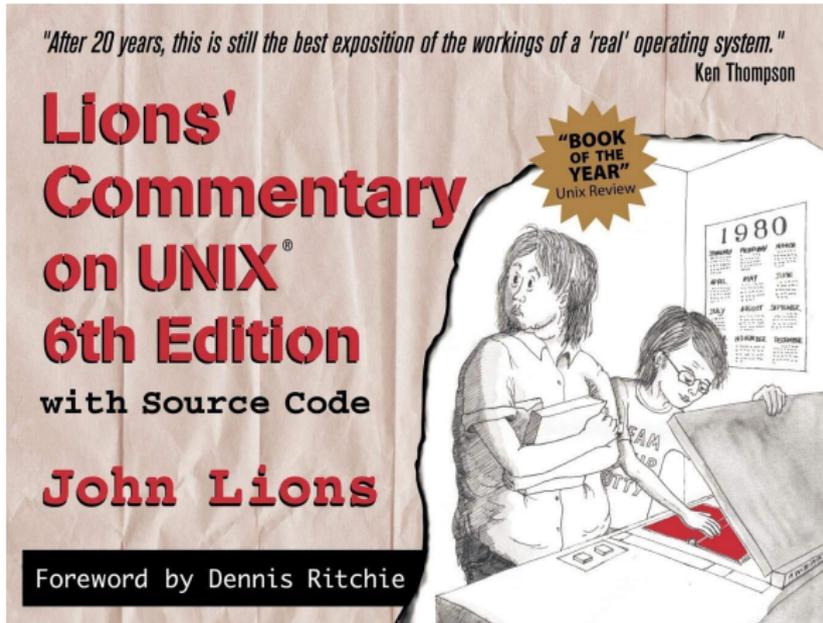
“Voici la philosophie d'Unix :

- Écrivez des programmes qui effectuent une seule chose et qui le font bien.
- Écrivez des programmes qui collaborent.
- Écrivez des programmes pour gérer des flux de texte, car c'est une interface universelle”

Il y a une page sur wikipedia ! https://en.wikipedia.org/wiki/Unix_philosophy

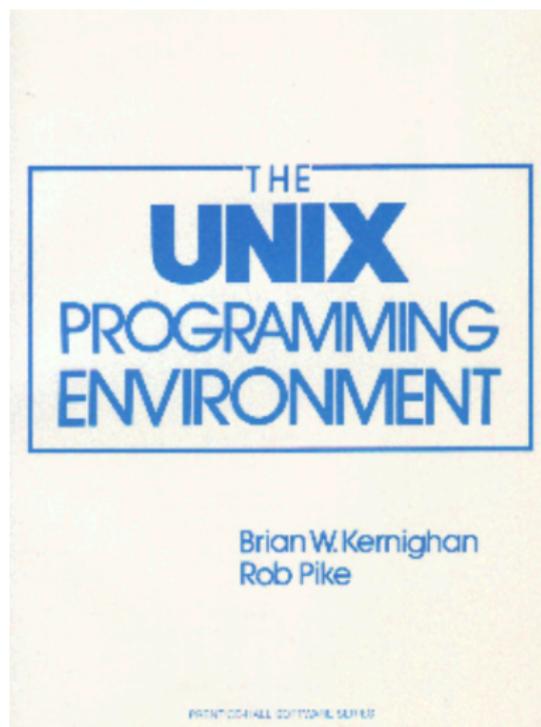
Littérature

Premier livre sur UNIX, 1976



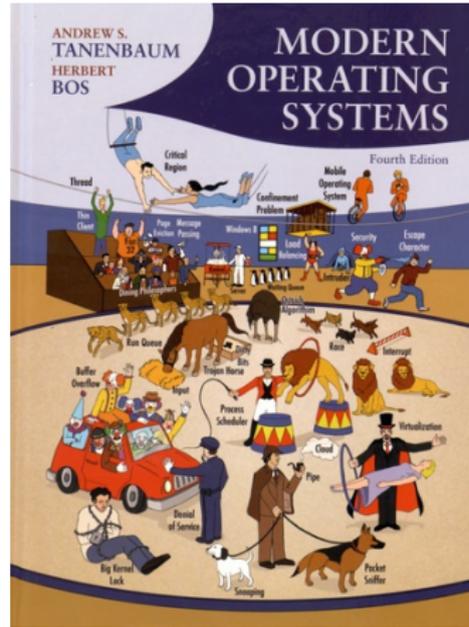
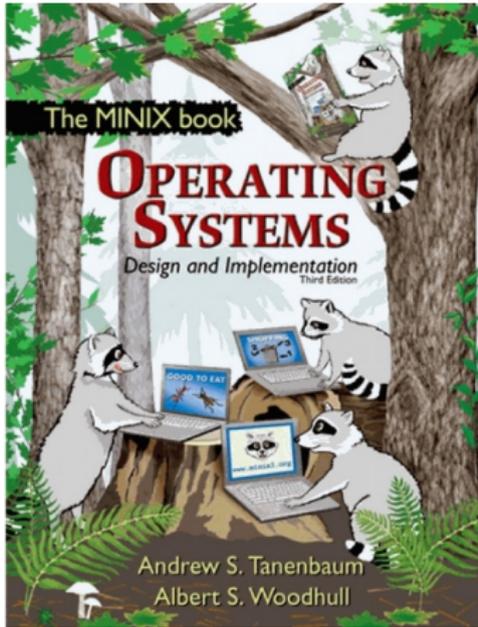
John Lions

Un autre bon vieux livre



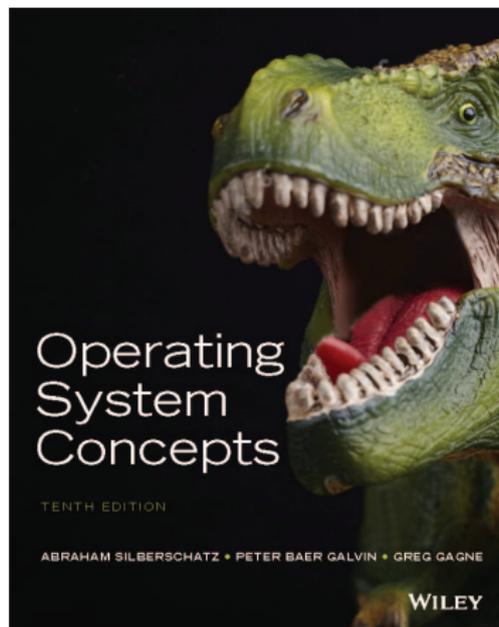
The Unix Programming Environment
Brian W. Kernighan and Rob Pike

MINIX, papa de Linux



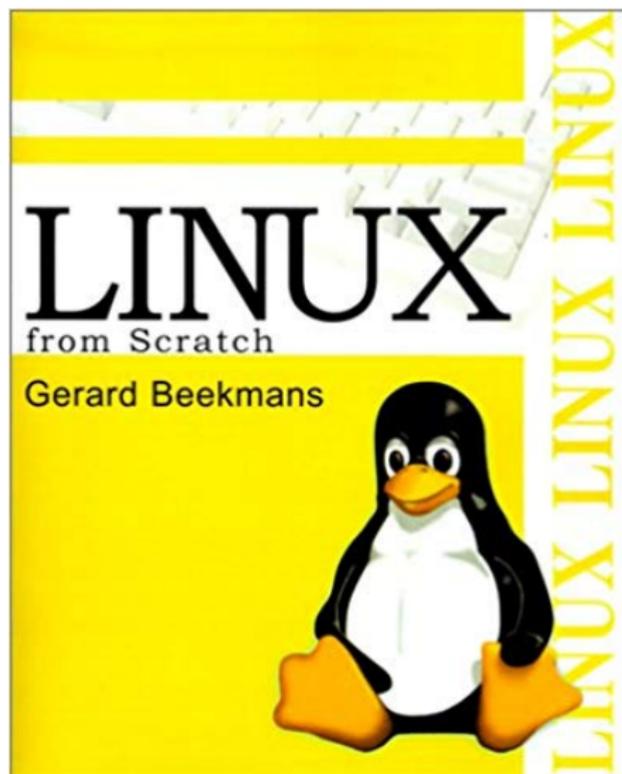
- Operating Systems : Design and Implementation.
Andrew S. Tanenbaum et Albert S. Woodhull.
- Modern Operating Systems.
Andrew S. Tanenbaum, Herbert Bos

Un autre bon livre sur les systèmes d'exploitation



<https://www.os-book.com/OS10/index.html> Operating systems concepts

Avi Silberschatz, Peter Baer Galvin, Greg Gagne



Le projet *Linux From Scratch* est un livre relatant les diverses étapes pour construire à partir des codes sources un système GNU/Linux.
Par Gerard Beekmans et al.



<http://9front.org>

Groupes de discussion existent sur IRC et Discord.

Architecture UNIX

UNIX aujourd'hui

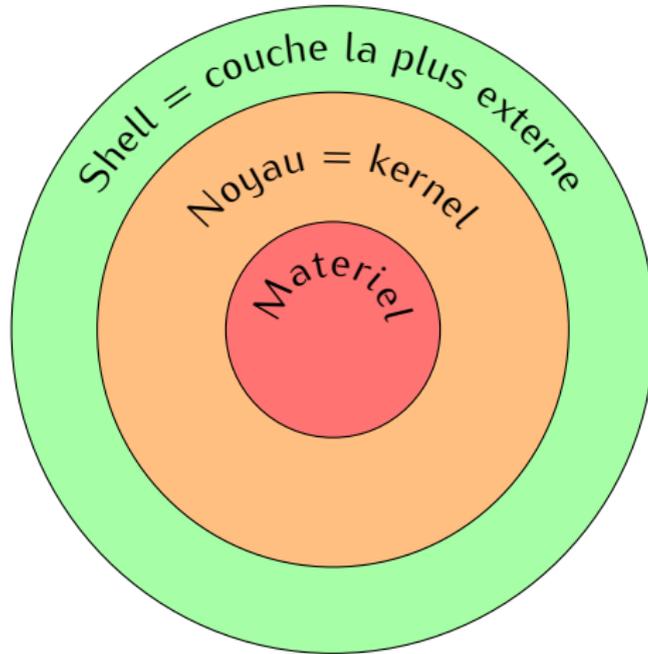
Présence d'UNIX / UNIX-like

- Linux depuis 1990
- Mac OS X : dérivé d'UNIX
- Solaris et Open Solaris
- Free BSD / Net BSD / Open BSD
- Et bien d'autres ...

La famille Linux

- Nombreuses distributions orientées particuliers
- Distribution = noyau Linux + packages spécifiques
- Interface graphique spécifiques, outils différents, outils de d'installation et de mise à jour des programmes différents, fréquence des maj, ...
- Mais globalement : même système d'exploitation

Utilisateurs



Au coeur du système : le noyau (kernel)

Le noyau : la partie principale du système d'exploitation

- Gestion des ressources
- Communication entre logiciel et matériel
- Niveau d'abstraction pour le matériel
- Communication inter-processus
- Gestion du système de fichiers
- Support réseau
- Modèle de pilotes pour le matériel

Comporte de nombreuses fonctions qui permettent un accès direct au matériel (disques durs, mémoire, ...)

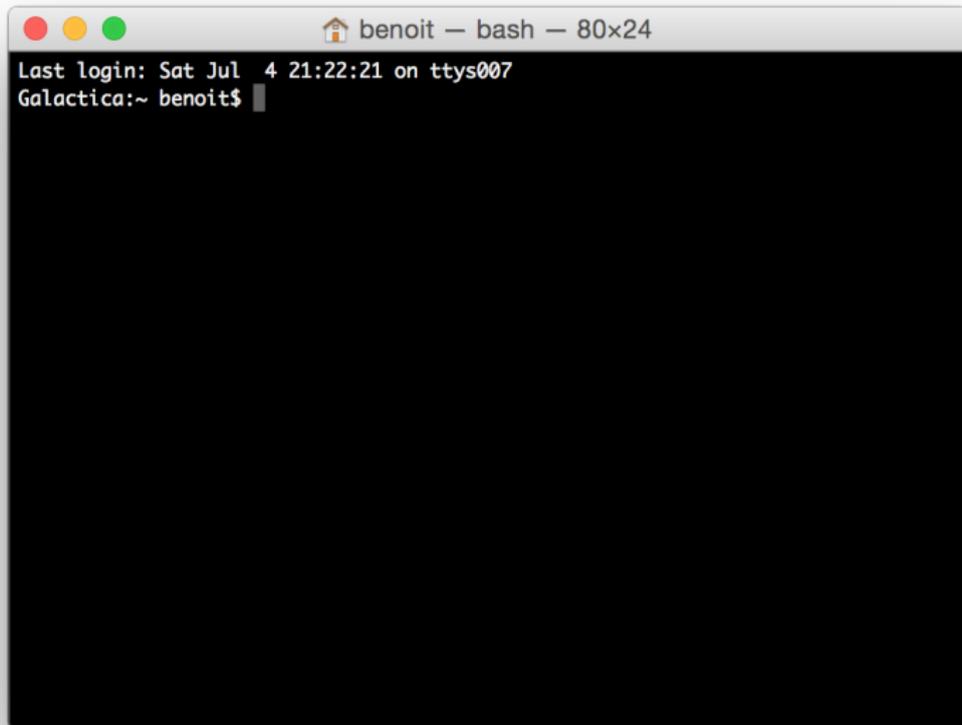
Le shell UNIX (anglais : coquille)

Le shell UNIX est un programme qui permet d'accéder aux fonctionnalités du système d'exploitation (1/2)

- Logiciel du système d'exploitation
- Interaction entre l'utilisateur et le système d'exploitation
- lancement / arrêt des programme, contrôle des processus, manipulation des fichiers, ...
- Interprétation et exécution des instructions de l'utilisateur :
Interpréteur de commandes
- Outil très puissant et `programmable` : manipulation de variables

Le shell se présente sous la forme d'une interface en ligne de commande accessible depuis la console à partir d'un terminal.

Exemple de shell UNIX lancé avec l'outil `terminal`
(équivalent windows : `cmd.exe`)



```
benoit — bash — 80x24
Last login: Sat Jul  4 21:22:21 on ttys007
Galactica:~ benoit$
```

Le shell UNIX

Le shell UNIX est un programme qui permet d'accéder aux fonctionnalités du système d'exploitation (2/2)

- Il n'existe pas qu'un seul shell, mais plusieurs possibles
- bash, tcsh, csh, sh, ...
- Mêmes fonctions générales
- Syntaxe, commandes internes, et comportements par défauts peuvent varier légèrement à modérément selon les shells
- Ligne de commande vs graphiques

Accès à un système UNIX

Utilisation d'UNIX

- UNIX est un système multi-utilisateurs
- Besoin d'un compte utilisateur pour pouvoir s'authentifier sur le système d'exploitation
- Une fois authentifié : possibilité de lancer des applications
- Différentes classes d'utilisateurs

Accès à un système UNIX

Utilisateurs du système

Utilisateurs standards

- Peuvent exécuter des applications, gérer un espace personnel
- Ne sont pas autorisés à installer / supprimer des applications
- Ne peuvent pas modifier les parties sensibles du système

Administrateurs

- Possèdent tous les droits d'un utilisateur standard
- Peuvent également modifier les parties sensibles du système : configuration, accès aux espaces personnels des utilisateurs, ...

Utilisateurs virtuels

- Dédiés à l'exécution de certains services et applications

Utilisateurs du système

Utilisateurs standards

- Peuvent exécuter des applications, gérer un espace personnel
- Ne sont pas autorisés à installer / supprimer des applications
- Ne peuvent pas modifier les parties sensibles du système

Administrateurs

- Possèdent tous les droits d'un utilisateur standard
- Peuvent également modifier les parties sensibles du système : configuration, accès aux espaces personnels des utilisateurs, ...

Utilisateurs virtuels

- Dédiés à l'exécution de certains services et applications

Utilisateurs du système

Utilisateurs standards

- Peuvent exécuter des applications, gérer un espace personnel
- Ne sont pas autorisés à installer / supprimer des applications
- Ne peuvent pas modifier les parties sensibles du système

Administrateurs

- Possèdent tous les droits d'un utilisateur standard
- Peuvent également modifier les parties sensibles du système : configuration, accès aux espaces personnels des utilisateurs, ...

Utilisateurs virtuels

- Dédiés à l'exécution de certains services et applications

Utilisateurs du système

Remarque

- Utilisateurs organisés en groupes
- Permet de gérer des accès en fonction de communautés
- Chaque utilisateur appartient à au moins un groupe

Utilisateurs du système

Utilisateurs UNIX caractérisés par sept éléments

- 1 **Identifiant**, ou login
- 2 **Mot de passe**
- 3 **UID** : User ID, ou numéro d'identifiant utilisateur
- 4 **GID** : Group ID, ou numéro d'identifiant du groupe par défaut
- 5 **Nom** : Nom réel de l'utilisateur
- 6 **Répertoire personnel** : Répertoire propre à l'utilisateur, dans lequel il peut stocker des fichiers.
- 7 **Shell** : Le premier programme à être lancé lorsque l'utilisateur s'authentifie sur le système.

Utilisation d'un système UNIX

Accès au système

- Saisie de l'identifiant puis du mot de passe
- En mode console le mot de passe n'est pas affiché, ni même les caractères * : tout taper à l'aveugle

Exécution de commandes

- Ouverture d'un terminal qui exécute un shell
- Saisie des commandes et exécution

Déconnexion du système

- Via l'interface graphique
- commandes `logout` ou `exit` en ligne de commande

Utilisation d'un système UNIX

Accès au système

- Saisie de l'identifiant puis du mot de passe
- En mode console le mot de passe n'est pas affiché, ni même les caractères * : tout taper à l'aveugle

Exécution de commandes

- Ouverture d'un terminal qui exécute un shell
- Saisie des commandes et exécution

Déconnexion du système

- Via l'interface graphique
- commandes `logout` ou `exit` en ligne de commande

Utilisation d'un système UNIX

Accès au système

- Saisie de l'identifiant puis du mot de passe
- En mode console le mot de passe n'est pas affiché, ni même les caractères * : tout taper à l'aveugle

Exécution de commandes

- Ouverture d'un terminal qui exécute un shell
- Saisie des commandes et exécution

Déconnexion du système

- Via l'interface graphique
- commandes `logout` ou `exit` en ligne de commande

Notion de commandes

Notion de commandes

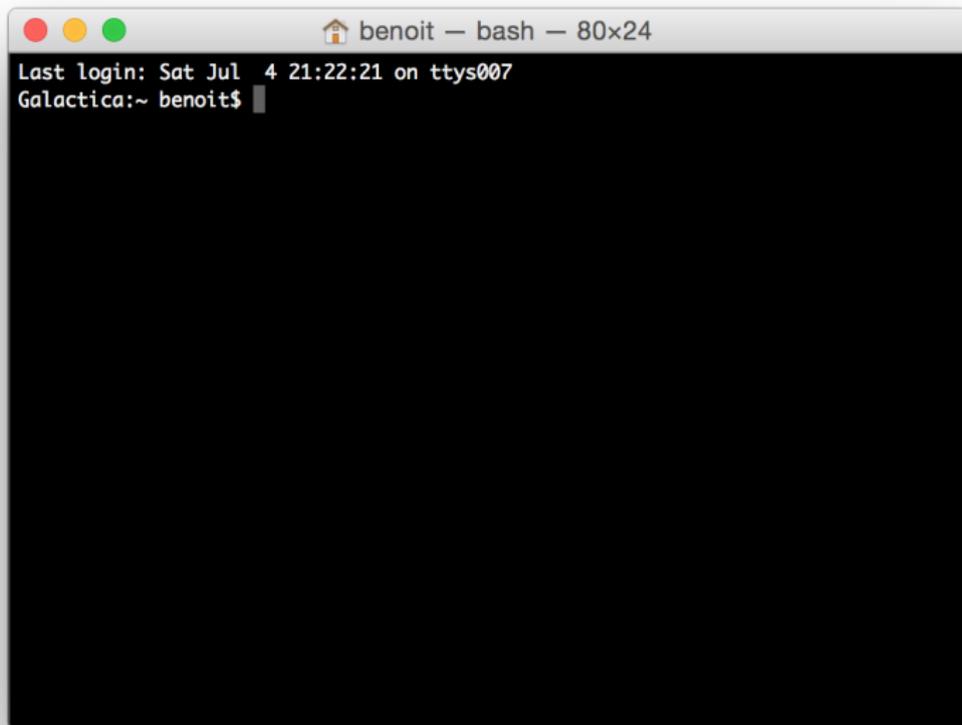
Une commande

- Programme : effectue une suite d'instruction
- Interaction entre l'utilisateur et le système d'exploitation
- Action précise et bien définie
- Commandes UNIX : agit sur le système, effectue une opération, renvoie un résultat,...

Syntaxe :

- Propre à chaque commande
- Commande simples
- Commande avec option et arguments
 - Commande : *quoi faire ?*
 - Option : *comment le faire ?*
 - Argument : *sur quoi le faire ?*

Saisie des commandes dans le shell (notez le prompt)

A terminal window with a title bar containing three colored buttons (red, yellow, green) and the text "benoit — bash — 80x24". The terminal content shows "Last login: Sat Jul 4 21:22:21 on ttys007" followed by the prompt "Galactica:~ benoit\$" with a cursor. The rest of the terminal area is black.

```
benoit — bash — 80x24
Last login: Sat Jul 4 21:22:21 on ttys007
Galactica:~ benoit$
```

Exécution de premières commandes

La commande `date`

- Affiche l'heure et le jour configurés sur le système

La commande `hostname`

- Affiche le nom de la machine à l'écran

La commande `echo`

- Affiche à l'écran un message passé en argument

```
1 benoit$ date
2 Lun 20 jul 2015 10:33:21 CEST
3
4 benoit$ hostname
5 Galactica
6
7 benoit$ echo "hello world"
8 hello world
```

Exécution de premières commandes

La commande `date`

- Affiche l'heure et le jour configurés sur le système

La commande `hostname`

- Affiche le nom de la machine à l'écran

La commande `echo`

- Affiche à l'écran un message passé en argument

```
1 benoit$ date
2 Lun 20 jul 2015 10:33:21 CEST
3
4 benoit$ hostname
5 Galactica
6
7 benoit$ echo "hello world"
8 hello world
```

Exécution de premières commandes

La commande `date`

- Affiche l'heure et le jour configurés sur le système

La commande `hostname`

- Affiche le nom de la machine à l'écran

La commande `echo`

- Affiche à l'écran un message passé en argument

```
1 benoit$ date
2 Lun 20 jul 2015 10:33:21 CEST
3
4 benoit$ hostname
5 Galactica
6
7 benoit$ echo "hello world"
8 hello world
```

Exécution de premières commandes

La commande `date`

- Affiche l'heure et le jour configurés sur le système

La commande `hostname`

- Affiche le nom de la machine à l'écran

La commande `echo`

- Affiche à l'écran un message passé en argument

```
1 benoit$ date
2 Lun 20 jul 2015 10:33:21 CEST
3
4 benoit$ hostname
5 Galactica
6
7 benoit$ echo "hello world"
8 hello world
```

Variables d'environnement

Variables et shell

On peut manipuler des variables dans le shell

- Création d'une variable par simple affectation : opérateur =
- syntaxe : `nomVar=valeur`
- Non typées, mais besoin d'un identificateur (nom) :
 - Ne peut pas commencer par un chiffre
 - Caractères autorisés : 0..9, A..Z, a..z, _
- Récupérer la valeur d'une variable : `$nomVar`
- Afficher à l'écran la valeur d'une variable : `echo $nomVar`

```
1 benoit$ A=bonjour
2 benoit$ B=5
3
4 benoit$ echo $A
5 bonjour
6 benoit$ echo A
7 A
8 benoit$ echo A vaut $A et B vaut $B
9 A vaut bonjour et B vaut 5
```

Variables existants

Certaines variables existent déjà.

Créées par le shell, mais accessibles à l'utilisateur.

Accès identique, mais leur valeur est gérée par le shell, par exemple

Exemples

Nom	Contenu
RANDOM	Génère un nombre aléatoire entre 0 et 32767
LOGNAME	Affiche le nom de l'utilisateur
HOSTNAME	Affiche le nom de la machine

Variables et shell

```
1 benoit$ echo $HOSTNAME
2 Galactica
3
4 benoit$ echo $LOGNAME
5 benoit
6
7 benoit$ echo $RANDOM
8 24845
9
10 benoit$ echo $RANDOM
11 29568
12
13 benoit$ echo $RANDOM
14 25607
15
16 benoit$ echo $RANDOM
17 18727
```

Question

- 1 Créez une variable nommée `prénom` et affectez-lui votre prénom
- 2 Créez une variable nommée `nom` et affectez-lui votre nom
- 3 Affichez un message de bienvenue personnalisé à votre prénom et nom

Question

- 1 Créez une variable nommée `prénom` et affectez-lui votre prénom
- 2 Créez une variable nommée `nom` et affectez-lui votre nom
- 3 Affichez un message de bienvenue personnalisé à votre prénom et nom

Réponse :

- 1 `prenom='benoit'`
- 2 `nom='darties'`
- 3 `echo "bonjour $prenom $nom. Comment allez-vous?"`

La documentation UNIX

Trouver de l'aide

Nombreuses sources de documentation

Tutoriels, forum de discussion, moteur de recherches, RFC, wiki, livres ...

Documentation incluse dans le système

- `man` : commandes externes, fonctions, programmes, ...
- `help` : commandes internes

Ces documents font office de référence dans la communauté UNIX

Merci.