

useradd



La commande `useradd` est une commande sous Linux utilisée pour créer un nouveau compte utilisateur sur le système. Elle est généralement utilisée par les administrateurs système pour ajouter de nouveaux utilisateurs, tels que des utilisateurs réguliers ou des comptes de service, au système.

La syntaxe de base de la commande `useradd` est la suivante :

```
useradd [options] nom_utilisateur
```

où `nom_utilisateur` est le nom du nouveau compte à créer, et `[options]` sont des options facultatives permettant de spécifier différents paramètres pour le nouveau compte. Parmi les options courantes, on trouve :

- `-c` commentaire : spécifie un commentaire pour le compte.
- `-m` : crée le répertoire personnel pour le compte s'il n'existe pas déjà.
- `-s` shell : spécifie le shell par défaut pour le compte.
- `-g` groupe : spécifie le groupe principal pour le compte.
- `-G` groupes : spécifie les groupes supplémentaires pour le compte (séparés par des virgules).

Par exemple, pour créer un nouveau compte utilisateur nommé `alice` avec le shell Bash et le groupe principal `users`, vous pouvez utiliser la commande suivante :

```
useradd -m -s /bin/bash -g users alice
```

Il est important de noter que la commande `useradd` crée un compte utilisateur, mais ne définit pas de mot de passe pour ce compte. Il est donc recommandé de définir un mot de passe pour le compte dès que possible après sa création avec la commande `passwd`.

Script de création de compte avancé

Voici un exemple de script Bash qui utilise la commande `useradd` pour créer un compte utilisateur nommé `alice` avec un mot de passe aléatoire et stocke l'adresse e-mail de l'utilisateur demandée au préalable, envoie un mail pour le mot de passe :

```
#!/bin/bash

# Définition des couleurs pour l'affichage de messages
RED='\033[0;31m'
GREEN='\033[0;32m'
```

```
YELLOW='\033[0;33m'
NC='\033[0m' # No Color

# Fonction pour afficher un message d'erreur
function print_error() {
    echo -e "${RED}[ERROR] $1${NC}"
}

# Fonction pour afficher un message d'information
function print_info() {
    echo -e "${YELLOW}[INFO] $1${NC}"
}

# Fonction pour afficher un message de succès
function print_success() {
    echo -e "${GREEN}[SUCCESS] $1${NC}"
}

# Fonction pour ajouter un nouvel utilisateur avec la commande useradd
function add_user() {
    local username=$1
    local realname=$2
    local homedir=$3
    local shell=$4
    local groups=$5
    local email=$6

    # Vérifier si l'utilisateur existe déjà
    if id -u "$username" >/dev/null 2>&1; then
        print_error "L'utilisateur $username existe déjà."
        return 1
    fi

    # Créer un nouveau répertoire home pour l'utilisateur
    if [ ! -d "$homedir" ]; then
        mkdir -p "$homedir"
        chmod 700 "$homedir"
        chown "$username:$username" "$homedir"
    fi

    # Générer un mot de passe aléatoire
    local password=$(pwgen -1 12)

    # Envoyer le mot de passe à l'adresse e-mail de secours si fournie
    if [ -n "$email" ]; then
        echo "Le mot de passe pour l'utilisateur $username est : $password" |
        mail -s "Mot de passe utilisateur" "$email"
    fi

    # Stocker l'adresse e-mail de secours
    echo "$email" > "$homedir/.email-secours"
```

```
# Ajouter l'utilisateur avec la commande useradd
useradd -c "$realname" -d "$homedir" -s "$shell" -G "$groups" -m
"$username"

# Définir le mot de passe de l'utilisateur
echo "$username:$password" | chpasswd

# Inviter l'utilisateur à changer de mot de passe au prochain démarrage
passwd -e $username

print_success "L'utilisateur $username a été ajouté avec succès."
}
```

Ce script génère un mot de passe avec la commande pwgen qui peut générer des mots de passe aléatoires sécurisés, utilise la commande useradd pour créer un nouveau compte utilisateur nommé alicia avec le mot de passe aléatoire et stocke l'adresse e-mail de secours de l'utilisateur dans le fichier .email-secours du répertoire de l'utilisateur, puis envoie le nouveau mot de passe par courrier électronique à l'adresse e-mail de l'utilisateur à l'aide de la commande mail.

Exemple d'utilisation de la fonction add_user :

```
add_user "alice" "Alice Doe" "/home/alice" "/bin/bash" "sudo,www-data"
"alice.doe@example.com"
```

From:

<https://www.abonnel.fr/> - notes informatique & technologie

Permanent link:

<https://www.abonnel.fr/informatique/linux/commandes/useradd>

Last update: **2023/02/19 11:12**

