

CENTRALESUPÉLEC

SYSTÈMES D'EXPLOITATION

---

*La programmation avec le Shell/Unix*

---

*Élève:*

Daniel STULBERG HUF

*Encadrant:*

Idir AIT SADOUNE

# Table des matières

<b>1</b>	<b>Sauvegarde automatique de fichiers</b>	<b>1</b>
<b>2</b>	<b>Gestion d'une Corbeil de fichiers</b>	<b>2</b>
<b>3</b>	<b>Compilation de fichiers Java</b>	<b>4</b>

# 1 Sauvegarde automatique de fichiers

## sauvegarder.sh

```
#!/bin/bash

sauvegarde_dir="OLD"

# Vérification de l'existence du répertoire OLD
if [ ! -d "$sauvegarde_dir" ]
then
    mkdir "$sauvegarde_dir"

    # Vérification de la création du répertoire OLD
    if [ $? -eq 0 ]
    then
        echo "Le répertoire $sauvegarde_dir a été créé avec succès !"
    else
        echo "Erreur lors de la création du répertoire de sauvegarde $sauvegarde_dir"
        exit 1
    fi
fi

# Obtention de la date courante
date=$(date +%Y-%m-%d)

# Parcours des fichiers du répertoire courant
for fichier in *
do
    # Vérification que le fichier est un fichier ordinaire
    if [ -f "$fichier" ]
    then
        # Création d'un nouveau nom de fichier avec la date courante
        nom="${fichier%.*}"
        extension="${fichier##*.}"
        nouveau_nom="$sauvegarde_dir"/"$nom#$date.$extension"

        # Copie du fichier dans le répertoire de sauvegarde
        cp "$fichier" "$nouveau_nom"

        if [ $? -eq 0 ]
        then
            echo "Le fichier '$fichier' a été copié avec succès dans le répertoire $sauvegarde_dir"
        else
            echo "Erreur lors de la copie du fichier '$fichier' dans le répertoire $sauvegarde_dir"
        fi
    fi
done

echo "Sauvegarde terminée !"
```

## 2 Gestion d'une Corbeil de fichiers

### jeter.sh

```
#!/bin/bash

poubelle_dir="$HOME/poubelle"

# Vérification de l'existence de la poubelle
if [ ! -d "$poubelle_dir" ]
then
    mkdir -p "$poubelle_dir"

    # Vérification de la création de la poubelle
    if [ $? -eq 0 ]
    then
        echo "La poubelle a été créée avec succès !"
    else
        echo "Erreur lors de la création de la poubelle"
        exit 1
    fi
fi

# Fonction pour lister le contenu de la poubelle
function lister_poubelle {
    echo "Contenu de la poubelle :"
    echo "-----"
    ls -l "$poubelle_dir"
}

# Fonction pour sortir un fichier de la poubelle
function sortir_poubelle {
    # Vérification du nombre d'arguments
    if [ $# -ne 2 ]
    then
        echo "Nombre d'arguments incorrect"
        echo "Usage : $0 -s <fichier> <chemin>"
        exit 1
    fi

    # Vérification de l'existence du fichier
    if [ ! -f "$poubelle_dir/$1" ]
    then
        echo "Le fichier '$1' n'existe pas dans la poubelle"
        exit 1
    fi

    # Demande de confirmation
    echo "Voulez-vous vraiment sortir le fichier '$1' de la poubelle ? (y/n)"
    read reponse

    if [ $reponse = "y" ]
    then
        mv "$poubelle_dir/$1" "$2"
        if [ $? -eq 0 ]
        then
            echo "Le fichier '$1' a été sorti de la poubelle"
        else
            echo "Erreur lors de la sortie du fichier '$1' de la poubelle"
        fi
    fi
}
```

```

else
    echo "Le fichier '$1' n'a pas été sorti de la poubelle"
fi
}

# Fonction pour vider la poubelle
function vider_poubelle {
    # Demande de confirmation
    echo "Voulez-vous vraiment vider la poubelle ? (y/n)"
    read reponse

    if [ $reponse = "y" ]
    then
        rm -rf "$poubelle_dir"/*
        if [ $? -eq 0 ]
        then
            echo "La poubelle a été vidée"
        else
            echo "Erreur lors de la vidage de la poubelle"
        fi
    else
        echo "La poubelle n'a pas été vidée"
    fi
}

# Vérifier les options de la commande
case $1 in
    "-l") lister_poubelle;;
    "-s") sortir_poubelle "$2" "$3";;
    "-v") vider_poubelle;;
    *)
        for fichier in "$@"
        do
            # Vérification de l'existence du fichier
            if [ -e "$fichier" ]
            then
                # Déplacement du fichier dans la poubelle
                mv "$fichier" "$poubelle_dir"
                if [ $? -eq 0 ]
                then
                    echo "Le fichier '$fichier' a été déplacé dans la poubelle"
                else
                    echo "Erreur lors du déplacement du fichier '$fichier' dans la poubelle"
                fi
            else
                echo "Le fichier '$fichier' n'existe pas"
            fi
        done
    ;;
esac

```

## 3 Compilation de fichiers Java

### javac\_test.sh

```
#!/bin/bash

# Vérification du nombre d'arguments
if [ $# -ne 1 ]
then
    echo "Entrée incorrecte : nombre d'arguments incorrect"
    echo "Usage : $0 <fichier.java>"
    exit 2
fi

# Récupération du nom du fichier Java passé en argument
fichier_java="$1"
fichier_class="${fichier_java%.java}.class"

# Vérification du fichier Java
if [ ! -f "$fichier_java" ]
then
    echo "Entrée incorrecte : le fichier '$fichier_java' n'existe pas"
    exit 2
fi

# Compilation du fichier Java
if javac "$fichier_java" 2> /dev/null
then
    echo "$fichier_java : compilé"
    exit 0
else
    echo "$fichier_java : non compilé"
    exit 1
fi
```

### java\_compilation.sh

```
#!/bin/bash

# Vérification que le répertoire est fourni
if [ $# -ne 1 ]
then
    echo "Nombre d'arguments incorrect"
    echo "Usage : $0 <répertoire>"
    exit 1
fi

source_dir="$1"
dest_dir="$source_dir/bin"

# Création le répertoire de destination s'il n'existe pas
mkdir -p "$dest_dir"

# Parcours des fichiers du répertoire source
for fichier in "$source_dir"/*.java
do
    # Exécuter javac_test.sh sur le fichier
    if ./javac_test.sh "$fichier"
    then
```

```

    # Récupération du nom du fichier Java
    nom_fichier="${fichier%.java}"

    # Déplacement du fichier compilé dans le répertoire de destination
    mv "$nom_fichier.class" "$dest_dir"
fi
done

```

## java\_execution.sh

```

#!/bin/bash

# Vérification que le répertoire est fourni
if [ $# -ne 1 ]
then
    echo "Nombre d'arguments incorrect"
    echo "Usage : $0 <répertoire>"
    exit 1
fi

source_dir="$1"
class_dir="$source_dir/bin"

# Vérification que le répertoire de destination existe
if [ ! -d "$class_dir" ]
then
    echo "Le répertoire $class_dir n'existe pas"
    exit 1
fi

# Parcours des fichiers .class dans le répertoire
cd "$class_dir"
for fichier_class in ./*.class
do
    # Récupération du nom du fichier sans l'extension .class
    nom_fichier=$(basename "$fichier_class" .class)

    # Vérification que le fichier .class contient une méthode de signature public static void main
    if javap -public "$nom_fichier" | grep -q "public static void main"
    then
        # Exécution du fichier .class
        echo "-----"
        echo "Exécution du fichier $nom_fichier"
        echo "-----"
        java "$nom_fichier"
    fi
done

```