



Consignes

Travail en binôme, à remettre avant le **21 janvier 2011** à 14h. Dans `/net/Remise/ASR1-USI`, il y a un sous-répertoire pour chaque groupe. Créez-y un répertoire à vos noms, lisible par vous-mêmes et le groupe `info_personnels` uniquement. Dans ce répertoire, vous placerez :

- le rapport **au format PDF** impérativement
- le(s) fichier(s) source(s) du script.

1 Sujet

Réaliser un script interactif pour gérer une petite bibliothèque. Le script est destiné au bibliothécaire qui pourra :

- enregistrer de nouveaux ouvrages
- noter l'emprunt d'un ouvrage par un usager
- noter le retour d'un ouvrage par un usager
- voir la liste des ouvrages empruntés par un usager
- etc.

Un début de script est fourni.

1.1 Fichiers de données

Les données sont contenues dans deux fichiers du sous-répertoire `EMPRUNTS` du bibliothécaire :

- le catalogue (liste des livres)
- la liste des emprunts.

1.2 Actions

Le script prévoit déjà une dizaine d'actions

Commandes :

```
AC afficher catalogue
AE afficher emprunts
AL ajouter livre
...
```

La commande `X` crée le répertoire de données avec les fichiers vides, elle ne sert en principe qu'une fois lors de l'installation. La commande `RT` remet les fichiers dans un état connu, elle

permet au développeur de faire des tests à partir de données correctes.

Au besoin vous ajouterez des actions.

2 Rapport

Le rapport que vous remettrez contiendra :

- Une présentation détaillée de ce que vous avez réalisé : fichiers, et liste des fonctionnalités.
- pour chaque fonctionnalité :
 - une description précise de son rôle
 - une illustration claire (avec capture d'écran) de l'effet de chaque action
 - une explication des "astuces" techniques employées.
- une liste des problèmes connus.

3 Normes de programmation

Le travail fourni par un programmeur compétent est reconnaissable au respect des normes de programmation, qui rendent son code "évident" :

- découpage en fonctions, organisation logique du code
- noms de fonctions = verbes d'action
- variables bien nommées
- commentaires pertinents, code documenté
- ...

Ceci s'applique aux scripts.

4 Annexe : code fourni

```
#!/bin/bash

# gestion des emprunts de livres
# dans une bibliothèque

# attention , affichage en jeu de caractères ISO-LATIN-1

# auteurs :
# groupe :

# -----

repertoire=~ /EMPRUNTS
catalogue=$repertoire/catalogue.txt
emprunts=$repertoire/emprunts.txt

# format du catalogue des livres :
# - un livre par ligne
# - champs reference , titre et auteur séparés par ":"
#
# format de la liste des emprunts
# un emprunt par ligne
# - champs référence du livre , date d'emprunt ,
#   et identifiant de l'emprunteur séparés par ":"
# - l'identifiant est le nom du compte
# - date au format AAAA-MM-JJ
#####

function reinitialiser
{
    echo "Effacer toutes les données o/n?"
    read reponse
    if [ $reponse = o ]
    then
        [ -d $repertoire ] || mkdir $repertoire
        rm -f $catalogue $emprunts
        touch $catalogue $emprunts
        echo "Ok. Données effacées."
    else
        echo "Abandon."
    fi
}

# -----

function recharger_pour_tests
{
    echo "Effacer toutes les données o/n?"
    read reponse
    if [ $reponse = o ]
    then
        [ -d $repertoire ] || mkdir $repertoire
        cat > $catalogue <<EOF
1:Fundamental Algorithms:Donald Knuth
2:Les bases de l'administration systeme:Aelen Frisch
3:Building Secure Servers:Michael D. Bauer
4:Servlets Java:Jason Hunter
5:CVS, configuration et mise en oeuvre:Fred Lepied
6:Java in a Nutshell:David Flanagan
EOF
        cat > $emprunts <<EOF
2:2010-11-03:mibillau
3:2010-12-01:fmekhald
5:2010-11-22:smoreaud
EOF
        echo "Ok. Données de test chargées"
    else
        echo Abandon.
    fi
}
```

```

    fi
}

#####
#
# améliorer la présentation (colonnes)
#
function afficher_catalogue
{
    echo "CATALOGUE_DES_LIVRES"
    while IFS=":" read ref titre auteur
    do
        printf "%s_%s\n\t\tpar_%s\n" $ref "$titre" "$auteur"
    done < $catalogue
}

# -----
#
# améliorez la fonction afficher_emprunt en utilisant
# le nom en clair de l'emprunteur
#
# essayez d'abord ceci
#   ldapsearch -x -h dagobah.iut.bx1 -b dc=iut,dc=bx1 cn=$LOGIN displayName
# et "nettoyez" le résultat avec grep, cut ...
#

function afficher_emprunts
{
    echo "LES_EMPRUNTS"
    while IFS=":" read ref date login
    do
        printf "%s_%s_%12s_\n" "$ref" "$date" "$login"
    done < $emprunts
}

# -----

function afficher_retards
{
    #
    echo "A_faire_..."
    # Affiche les livres qui ont été empruntés il y a plus de 3 semaines.
    #
    # commande à étudier :
    #   date -d "+3 days" +"dans 3 jours on sera le %d du mois de %B"
    #
}

# -----

function afficher_lecteur
{
    #
    echo "A_faire_..."
    # demande l'identifiant
    #
}

# -----

function enregistrer_emprunt
{
    #
    echo "a_faire"
    # demande l'identifiant de l'emprunteur
    # vérifie qu'il existe
    # et la référence du livre
    # vérifie que le livre n'est pas déjà emprunté
    # demande confirmation
    # enregistre l'emprunt
    #
}

# -----

function enregistrer_retour
{

```

```

    echo "a_la_faire"
    # demande la reference du livre
    # verifie qu'il a bien été emprunté
    # affiche le nom de l'emprunteur
    # demande confirmation
    # efface l'emprunt
}

# Quitter l'application
function quitter
{
    # demande confirmation
    # si oui, exit
    exit 0
}

# -----

function afficher_menu
{
    cat <<EOF
    Commandes:
        AC afficher catalogue      AE afficher emprunts
        AL ajouter livre           L afficher infos lecteur
        AR afficher retards        R retour d'un livre
        E emprunt d'un livre       M afficher le menu
        Q quitter                  RT recharger données de test
        X reinitialiser
    EOF
}

# -----

function dialoguer
{
    cat <<EOF
    *****
    * Bibliotheque *
    *****
    EOF

    afficher_menu
    while true
    do
        read -p "Commande:_" reponse
        case "$reponse" in
            AC) afficher_catalogue ;;
            AL) ajouter_livre ;;
            AE) afficher_emprunts ;;
            AR) afficher_retards ;;
            L) afficher_lecteur ;;
            E) enregister_emprunt ;;
            R) enregister_retour ;;
            Q) quitter ;;
            X) reinitialiser ;;
            RT) recharger_pour_tests ;;
            M) afficher_menu ;;
            *) echo "Commande_inconnue."
               echo "tapez _M_ pour _voir_ le _menu"
               ;;
        esac
    done
}

# -----

dialoguer

# ----- fin -----

```