

Consignes

Le travail est à faire en binôme et à rendre **avant le 05 décembre 2021 23h50, dernier délai** sur la plateforme d'enseignement (par **un seul** membre du binôme) sous la forme d'un fichier `.tar.gz` contenant **uniquement** les deux fichiers suivants :

- Le script Shell nommé `convertisseur.sh` :
 - avec le nom des auteurs et la licence au début du fichier ;
 - aéré et indenté ;
 - écrit en utilisant des fonctions ;
 - commenté (dans le code) et documenté (en signature de fonctions).
- Le rapport **PDF** qui précise :
 1. les auteurs du travail ;
 2. ce que fait le script ;
 3. la syntaxe d'appel du script ;
 4. des traces d'exécution ;
 5. les difficultés rencontrées lors du projet ;
 6. le travail qui a été réalisé ;
 7. les améliorations éventuellement faites.
- Toute remise hors délais ne sera pas corrigée (la note est de 0).
- Tout rendu fait en monôme ou en trinôme ne sera pas corrigé (la note est de 0).
- Le plagia sera sévèrement sanctionné (la note est de 0 aussi bien pour le binôme source que le binôme destination et la procédure administrative adéquate est enclenchée).
- Le strict respect des consignes sera pris en compte dans la notation (vous codez au minimum ce que l'on demande, pas ce que vous voulez faire).

Les semaines de travail encadré sur ce projet sont celles du 08/11/2021, 15/11/2021, 22/11/2021. Vous avez également la semaine 29/11/2021 si besoin.

1 Présentation du sujet : le questionnaire de présentation HTML

On dispose d'enregistrements sonores d'un cours ou d'une conférence, et des fichiers PDF projetés pendant la présentation.

On veut présenter cette conférence sur le Web, sous forme d'une suite de pages HTML qui présentent chacune :

- un titre ;
- une page du fichier PDF ;
- un extrait sonore correspondant à cette page ;
- des boutons pour avancer/reculer vers la page suivante/précédente.

Une page sommaire doit référencer chacune des pages du document : cliquer sur le titre d'un document, ouvre la page associée. Pour cela on a constitué un «storyboard», sous forme d'un fichier qui a, sur chaque ligne, la description d'une page :

- le titre de la page
- le nom d'un fichier PDF
- le numéro de la page dans le PDF
- le nom de la piste sonore
- le début et la durée de l'extrait sonore ('minutes :secondes.centimes')

Exemple

```
Un,pres.pdf,1,exemple.amr,0:4.78,0:4.11
Deux,pres.pdf,2,exemple.amr,0:11,0:5
Trois,pres.pdf,3,exemple.amr,0:19,0:15
Quatre,pres.pdf,4,exemple.amr,0:37,0:9
Cinq,pres.pdf,4,exemple.amr,0:46,0:6
```

Note : le «magnétophone» d'un smartphone sous Android produit des fichiers AMR (environ 1,5 Ko par seconde) d'une qualité «téléphonique» suffisante pour une présentation.

En sortie, on produit un répertoire contenant les fichiers nécessaires à l'affichage correct de la présentation dans un navigateur web :

- images correspondants aux pages des PDF
- extraits sonores (formats OGG et MPG pour être compatibles)
- pages HTML. Ci dessous, un extrait de la page 3, qui a été codée en HTML 5 :

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Trois</title>
</head>
<body bgcolor="yellow">
  <h1>Trois</h1>

  
  <br/>
  <audio controls>
    <source src="exemple.amr-3.ogg" type="audio/ogg">
    <source src="exemple.amr-3.mp3" type="audio/mpeg">
    Your browser does not support the audio element.
  </audio>
  <a href="page-2.html">précédent</a>
  <a href="page-4.html">suivant</a>
</body>
</html>
```

2 Contraintes

- L'interface de l'application est la suivante :

```
$ ./convertisseur.sh -h
```

Usage:

```
./convertisseur.sh -f <source> -t <destination> [-s]
```

```
./convertisseur.sh -h
```

- -f <source> permet de spécifier le storyboard.
- -t <destination> permet de spécifier le dossier de destination.
- -s indique que nous souhaitons ouvrir la 1ere page dans le navigateur par défaut de l'utilisateur.

- `-h` Affiche l'aide d'utilisation.
- Le script doit être composé de plusieurs fonctions ayant chacune une petite responsabilité claire. Il doit y avoir entre autre une fonction `main` qui lance le programme et une fonction `usage` qui explique comment utiliser l'application.
- La langue anglaise doit être utilisée pour nommer les variables/fonctions et rédiger les commentaires.
- Les différents messages d'erreurs doivent être affichés sur la sortie d'erreur.
- Si le dossier de destination existe, l'application doit demander à l'utilisateur s'il veut le supprimer. Si oui, l'application supprime le dossier et lance la génération. Si non, l'application échoue.
- `shellcheck` indique que le code a très peu d'erreurs.

3 Quelques commandes utiles pour le projet

On peut extraire les pages du PDF avec `pdftoppm` et convertir le résultat en image avec `pnmtopng`. L'utilitaire `sox` permet de faire l'extraction (avec option `trim`) du son au format OGG que `ffmpeg` peut convertir en MP3.

Le début du script shell doit contenir les commandes suivantes

```
#!/bin/bash

set -o errexit # Exit if command failed.
set -o pipefail # Exit if pipe failed.
set -o nounset # Exit if variable not set.

# Remove the initial space and instead use '\n'.
IFS=$'\n\t'
```

4 Livrables

Dans la terminologie de la gestion de projet, les *livrables* sont les résultats concrets attendus du projet.

Le rapport contiendra

1. une notice rédigée à l'attention des utilisateurs, expliquant l'utilisation de cette commande
2. une page de manuel succincte
3. une notice de documentation à l'usage des programmeurs qui seront ultérieurement chargés de maintenir ce script, avec
 - principes généraux de fonctionnement du programme
 - fichiers et structures de données
 - découpage en fonctions
 - scénarios de test
 - ...
4. un compte-rendu de test, dans lequel vous préciserez les scénarios qui conduisent à des erreurs non encore résolues.

Le code source que vous joindrez évidemment sera rédigé avec soin, en veillant particulièrement

- à la structure (découpage en fonctions)
- à la documentation interne (commentaires précisant le rôle exact de chaque fonction, ses paramètres etc)
- au bon choix des identificateurs, etc.

La gestion des arguments peut être faite manuellement comme vous avez vu en cours ou automatiquement avec la fonction shell `getopts`.

Important : le script doit fonctionner sur les machines du département, sans avoir à installer d'autres programmes.