

## 1 Consignes

Le travail est à faire en **binôme** et à rendre **avant** le **21/01/2017 à 23h50** 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 :

1. Le script Shell nommé `quitter.sh` :
  - avec le nom des auteurs et la licence dans le cartouche du fichier ;
  - aéré et indenté ;
  - écrit en utilisant des fonctions ;
  - commenté et documenté (présence d'un cartouche pour chaque fonction, et code facile à comprendre).
2. 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 (21/01/2017 à 23h50)** ne sera pas corrigée (**la note est de 0**).
  - Tout rendu fait en monôme ou trinôme ne sera pas corrigé (**la note est de 0**).
  - 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).
  - Vérifiez immédiatement que vous avez accès à la plateforme d'enseignement (les clés sont 2017A, 2017B, 2017C, 2017D).

Les semaines de travail sur ce projet sont celles du 18/12/2016, 08/01/2017 et du 15/01/2017.

## 2 Présentation du sujet : le gestionnaire d'évènements

On ne voit pas le temps passer quand on est assis devant son ordinateur, et on oublie parfois des rendez-vous, des réunions, etc. On vous demande d'écrire un utilitaire qui servira à noter les rendez-vous, et prévient quand il est temps de partir, en rappelant la raison.

Scénario d'utilisation :

```
kevin@pc33:~$ quitter 1225 Resto avec Camille
kevin@pc33:~$ quitter 1900 Ciné Massacre à la tronçonneuse +cine +potes
...
```

à 12h25, une fenêtre s'ouvrira sur l'écran pour rappeler le rendez-vous. Idem à 19h00.

### 3 Quelques détails

- Les rendez-vous seront notés dans un fichier `horaires.db` situés dans le répertoire `~/.config/quitte` de l'utilisateur. Ce répertoire sera créé automatiquement.
- Sous la forme `quitte HHMM message... [+tag ...]` la commande `quitte` ajoute une ligne au fichier `horaires.db` et démarre en arrière-plan une boucle qui, à intervalle régulier (30 secondes par exemple), compare l'heure avec celle des événements du fichier et affiche le message de ceux qui correspondent (sans les tags). Les tags permettent d'optionnellement donner de la sémantique aux événements.
- On évitera de lancer plusieurs boucles qui risqueraient d'interférer. Pour cela on notera le numéro de processus de la boucle dans un fichier `boucle.pid` (toujours dans le dossier `~/.config/quitte`).
- L'application gère la cohérence des dates (impossibilité d'ajouter un événement dans le passé) et leur formatage (la date est toujours formatée de façon valide).
- Quelques options supplémentaires semblent utiles :
  - `quitte -q` arrêtera la boucle
  - `quitte -l [+tag ...]` affichera la liste des rendez-vous (l'heure et le message) à venir. Optionnellement on peut choisir de n'afficher que les événements qui correspondent à un tag particulier.
  - `quitte -a [+tag ...]` affichera la liste de tous les rendez-vous. Optionnellement on peut choisir de n'afficher que les événements qui correspondent à un tag particulier.
  - `quitte -r HHMM` supprime les rendez-vous à l'heure correspondante.
  - `quitte -r +tag` supprime les rendez-vous du tag correspondant.
  - `quitte -h` affiche l'aide de l'application.

### 4 Quelques éléments techniques

- Voir `xmessage` pour l'affichage.
- `date +%H%M` fournit l'heure courante sous la forme HHMM.
- La variable spéciale bash `$!` retourne le numéro du dernier processus lancé en arrière-plan.
- La variable spéciale bash `$?` retourne le code de retour du dernier processus lancé.

### 5 Éléments optionnels

Des éléments optionnels peuvent être ajoutés à partir du moment où tout le reste fonctionne. Sinon, ils ne seront pas pris en compte dans la notation.

- Possibilité de modifier le contenu d'un message.
- Gestion des événements sur plusieurs jours.
- Ajout de la possibilité d'exécuter une commande spécifique à l'événement en plus d'afficher un message.
- Configuration bash pour proposer l'autocomplétion des arguments du script.