

---

# **TD 11- Les scripts et les jobs**

Inès de Courchelle



2023-2024

**Objectifs :**

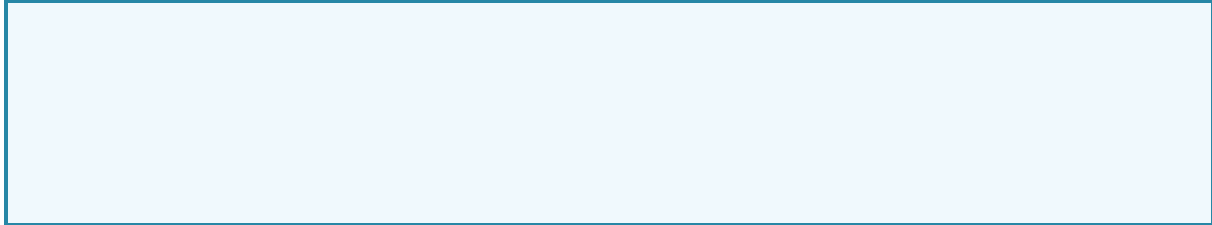
- Appréhender les modes d'exécution background et foreground
- Automatiser des commandes à l'aide de scripts
- Comprendre les différences entre les quotes
- Manipuler les arguments d'un script
- Utiliser toutes les commandes vues avant !

**Durée** 3h**Format** papier/ordinateur**Attention** Durant ce TD l'utilisation de la souris, Touchpad et/ou Trackpoint sont strictement interdites. Seul le clavier est autorisé.**Exercice 1 : Mon premier script**

1. Télécharger depuis le site le script suivant, `script1.sh`, et exécuter `script1.sh`. Que fait-il ?

2. Télécharger le script suivant, `script2.sh`. Que fait le script suivant ? Décrivez le programme ligne par ligne.

3. Exécutez un shell, qui affiche "bonjour" directement dans le terminal sans passer par un éditeur de texte.



## Exercice 2 : Mon premier script

Ecrire un programme, appelé `petitScript.sh` qui prend 5 paramètres/arguments et qui affiche :

- Son nom
- Son nombre de paramètres/arguments
- Le numéro de processus courant (celui qui exécute le script shell)
- La liste de paramètres que vous lui avez passés

### Quelques indices de variables utiles :

- `$0` ⇒ contient le nom du programme en cours
- `$1 .. $9` ⇒ contiennent les différentes valeurs des arguments/paramètres
- `$#` ⇒ contient le nombre de paramètres/arguments
- `$` ⇒ contient une liste de l'ensemble des paramètres/arguments
- `$?` ⇒ contient l'état de sortie de la dernière commande. Si la commande s'est bien passée, alors elle renvoie 0, sinon une valeur différente de zéro.
- `$$` ⇒ contient le numéro du processus en cours

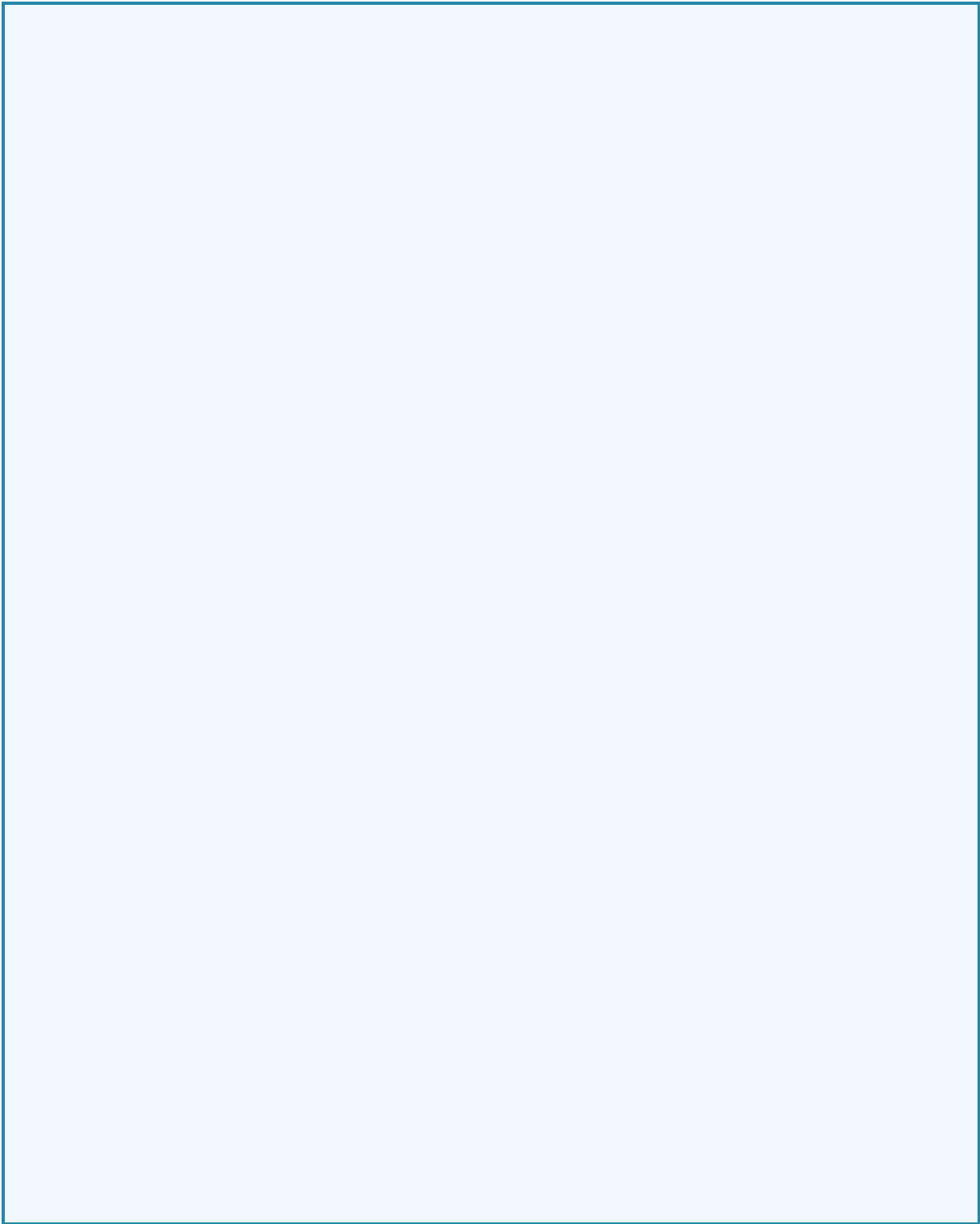


### Exercice 3 : Mon troisième script

1. Créer dans le répertoire de connexion un répertoire `monTroisième` qui contient les fichiers suivants (qui peuvent être vides) : *Attention : réaliser cette question depuis le terminal directement (inutile de créer un script permettant la création de ses fichiers)*

- `p1.c`
- `p2.pas`
- `p3.py`

- memo
  - p4.c
  - p5.txt
  - p6.sh
  - p7.sh
2. Depuis le terminal, rajouter manuellement le droit d'exécution à quelques fichiers, par exemple à : p3.py, p6.sh et p7.sh. **Rappel** : chmod
  3. Créer dans votre répertoire de connexion un script `maTache.sh`, qui doit :
    - être appelé avec un et un seul paramètre/argument
    - vérifier que le paramètre/argument est bien un répertoire
  4. Ajouter au script `maTache.sh` la capacité de consulter tous les fichiers du répertoire passé en paramètre et afficher le nom de ceux qui sont exécutables. Pour tester, il faut passer en paramètre le répertoire `monTroisième`. **Attention** un fichier en unix est dit exécutable s'il possède les droits d'exécution.
  5. Modifier le script pour qu'il affiche la taille de chacun des fichiers exécutables.
  6. Modifier le script pour qu'il compte le nombre des fichiers exécutables
  7. Modifier le script pour créer un répertoire `exec` et déplacer l'ensemble des exécutables à l'intérieur.



**Exercice 4 : Les modes d'exécution des processus**

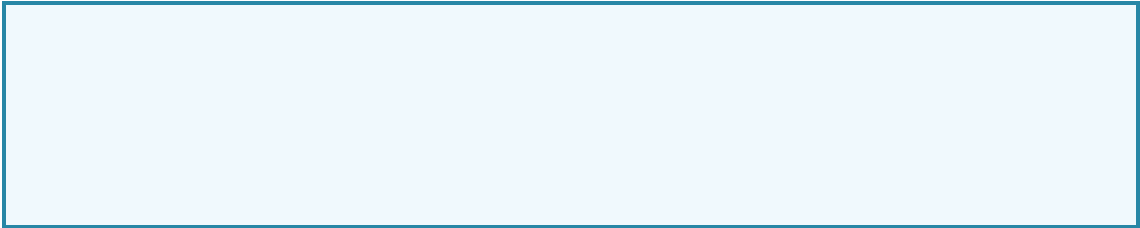
1. Créer un "script" qui effectue une boucle infinie contenant l'instruction `sleep`. **indices** : consulter le manuel de `sleep`

2. Exécuter le script en mode foreground. Est-il possible d'exécuter encore des commandes dans le terminal dans lequel le script a été lancé ?

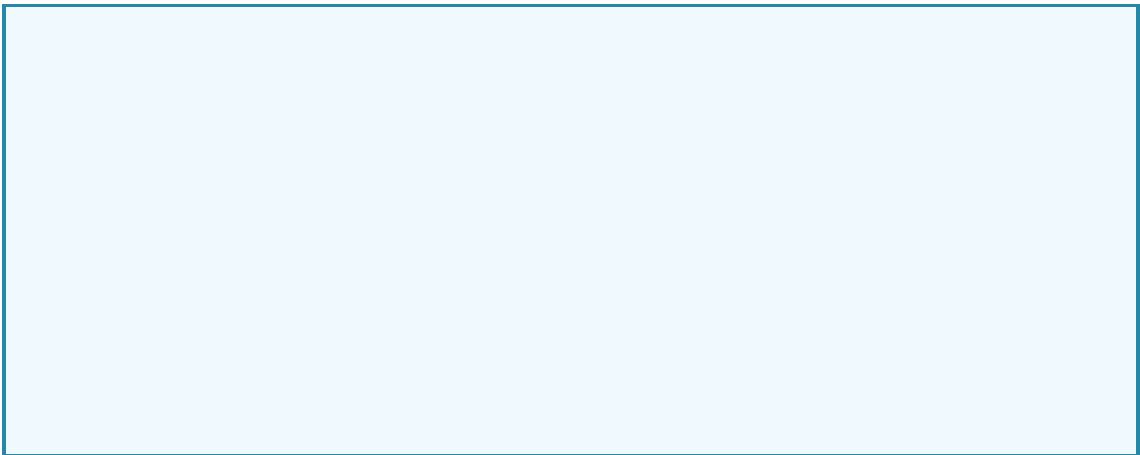
3. Arrêter le script, puis vérifier que le script n'est plus dans la liste des processus en cours.

4. Exécuter le script de manière à pouvoir utiliser le terminal.

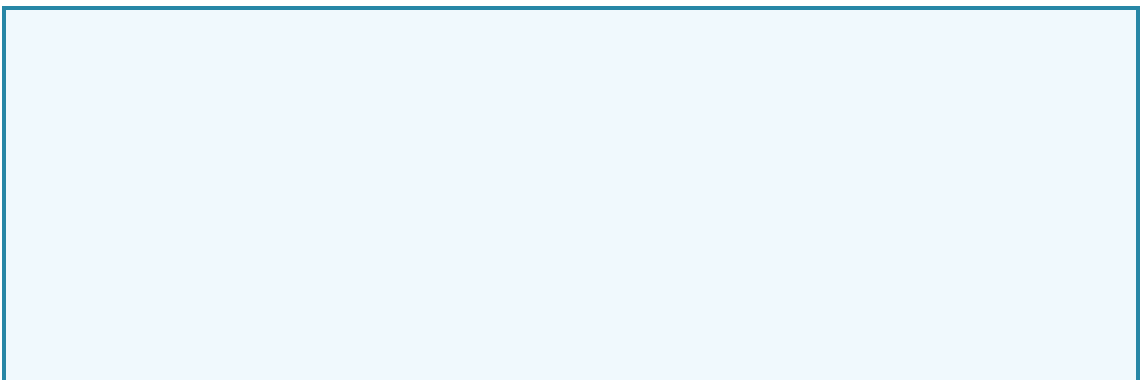
5. Exécuter le script en mode foreground puis passer le processus en mode background.



6. Exécuter en background le script, puis rechercher le dans la liste des processus en cours, et enfin, tuer le script.



7. Exécuter 2 script d'affiler en mode background (&), et tuer les deux processus.



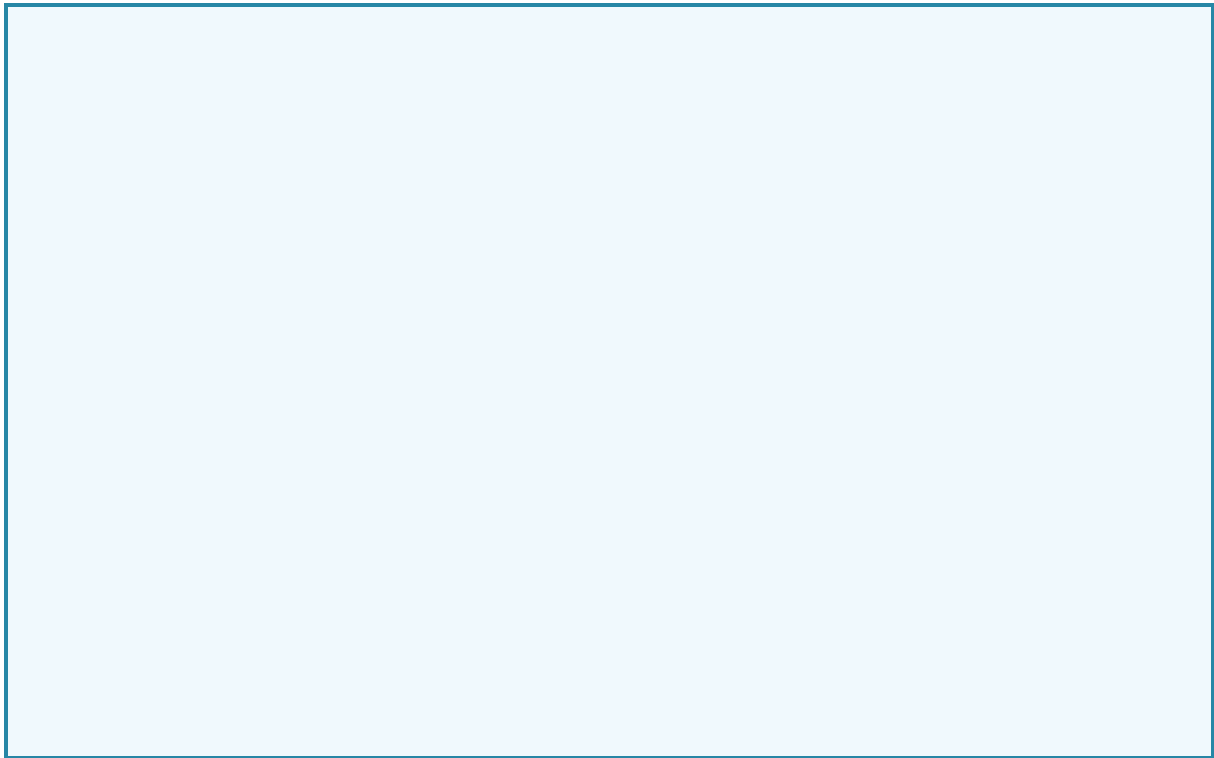


**Exercice 5 : Les révisions**

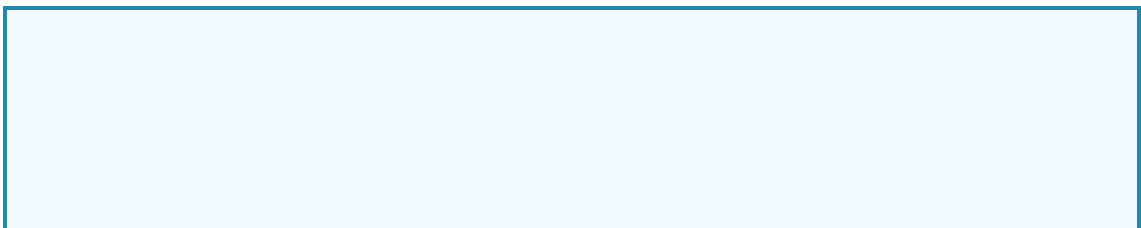
1. Nous considérons le fichier de notes suivant :

```
1 Liste de notes EISTI 2017
2 paco 10
3 pepe 5
4 jose 18
5 juan 15
6 luis 19.5
7 toto 2
8 titi 5
```

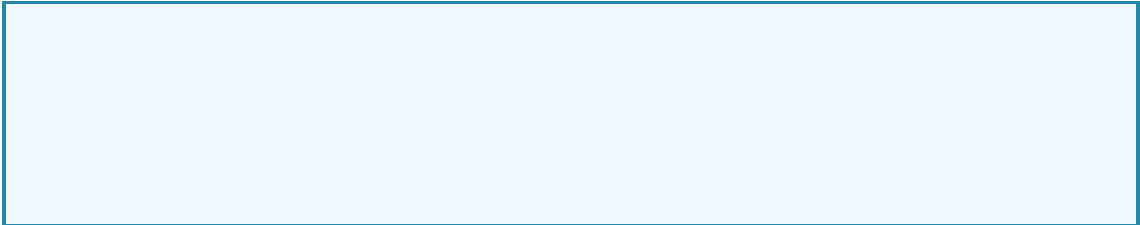
Créez un script qui permet un affichage trié d'un fichier de note sans trier l'entête (première ligne)



2. Affichez la liste des jobs dont vous êtes le propriétaires

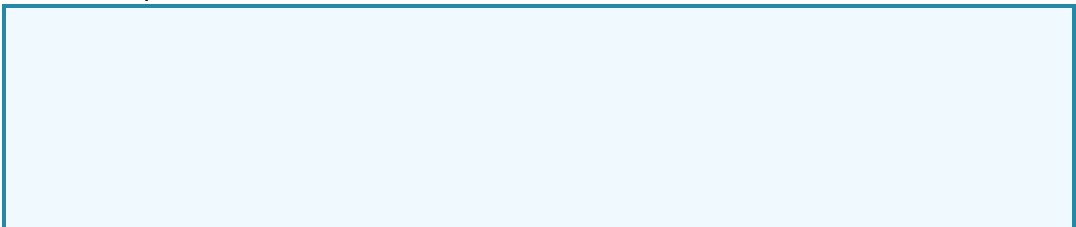


3. Affichez dans le terminal l'identifiant local des processus lancés

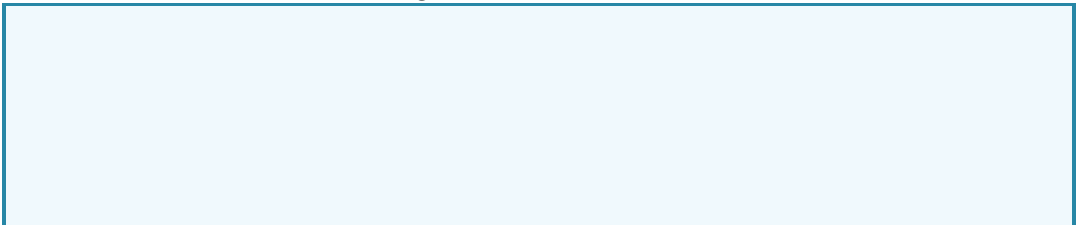


4. Nous considérons un script `toto.sh` :

1. Exécutez le processus



2. Passez le processus en mode background



3. Arrêtez le processus `toto.sh`

