

# ISS : Initiation au Système et au Shell

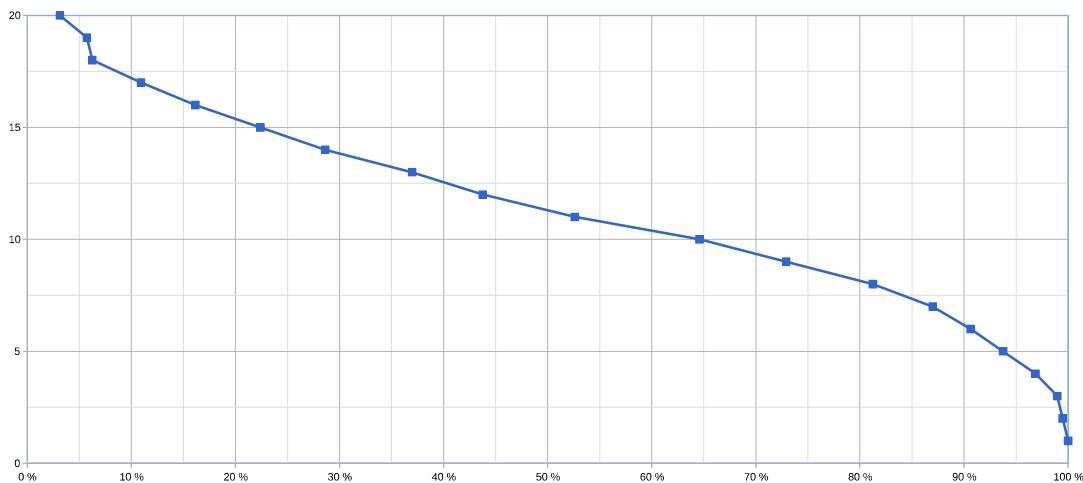
[Tableau de bord](#) / Mes cours / [LU2IN020 - ISS](#)

## Annonces

Les notes du partielle 2022 ont été publiées sur *dbufr*. Comme une note sans contexte n'est pas très parlante, voici un graphique qui vous permettra de vous situer.

Ce dernier donne la plus mauvaise note des meilleurs x % des étudiants. La médiane et le décile supérieur correspondent donc respectivement au point d'abscisse 50% et 10%.

Plus particulièrement, en cherchant le point dont l'ordonnée correspond à votre note, vous pourrez lire en abscisse votre position parmi les étudiants qui ont passé le partielle 2022 d'ISS.



## Règles de fonctionnement de l'UE

### **Modalités du contrôle des connaissances :**

- La note finale se décompose en : 50% pour la note de contrôle continu et 50% pour la note d'examen
- La note de contrôle continu se décompose en : 20% pour une note d'évaluation en TP et 80% pour la note du partielle
- Toutes les épreuves écrites se feront **sans documents** (voir les [consignes](#))

### **Consignes pour l'envoi de mails :**

- *Contenu ASCII* : pas de HTML, pas de fioritures ...
- *Sujet du mail* : commence par [ISS] suivi d'un sujet clair
- *Signature* : avec votre nom et votre numéro d'étudiant

## Équipe pédagogique

### **Cours**

[Julien Sopena](#), jeudi 16h-17h45, Amphi 44

### **Groupes de TD/TME**



Groupe	Enseignant	Créneau TD	Salle TD	Créneau TME	Salle TME
1	<a href="#">Jonathan Lejeune</a>	Lundi 10h45-12h30	24/25 Salle 103	Lundi 14h-15h45	14/15 Salle 308
2	<a href="#">Gabriel Le Bouder</a>	Mardi 10h45-12h30	24/25 Salle 106	Mardi 14h-15h45	24/25 Salle 304
3	<a href="#">Yvan Guifo</a>	Vendredi 10h45-12h30	23/24 Salle 102	Vendredi 14h-15h45	24/25 Salle 304
4	<a href="#">Aymeric Agon</a>	Jeudi 14h-15h45	23/24 Salle 107	Vendredi 14h-15h45	14/15 Salle 307
5	<a href="#">Alexandre Lavigne</a>	Mercredi 8h45-10h30	14/24 Salle 210	Mercredi 10h45-12h30	14/15 Salle 302
6	<a href="#">Franck Petit</a>	Jeudi 10h45-12h30	23/24 Salle 202	Jeudi 14h-15h45	24/25 Salle 302
DANT	<a href="#">Baptiste Pires</a>	Mercredi 8h45-10h30	14/24 Salle 316	Mercredi 10h45-12h30	14/15 Salle 406

## Consignes et aide technique pour les TP

### Pour les rendus de TP

Dans l'exercice 2 du TP-01, vous avez créé une arborescence de travail contenant un dossier pour chaque TP du semestre. Ce sont ces dossiers qui seront à rendre pour chacun des TPs.

Chaque rendu devra contenir à sa racine un fichier README, créé par vos soins et contenant les réponses aux différentes questions posées dans le sujet du TP. Ces réponses ne peuvent se faire sous la forme de screenshots du terminal, ni de fichiers Word. Tout doit être contenu dans le fichier texte README.

Vous êtes libre d'utiliser la convention que vous désirez pour numérotter les réponses, mais celles-ci doivent être dans l'ordre du sujet. Par exemple, vous pouvez simplement écrire le numéro des exos et des questions en texte brut (comme dans le screencast ci-dessous). Pour celles et ceux qui veulent aller plus loin, [Markdown](#) est le format le plus répondu pour rédiger le README d'un projet logiciel.

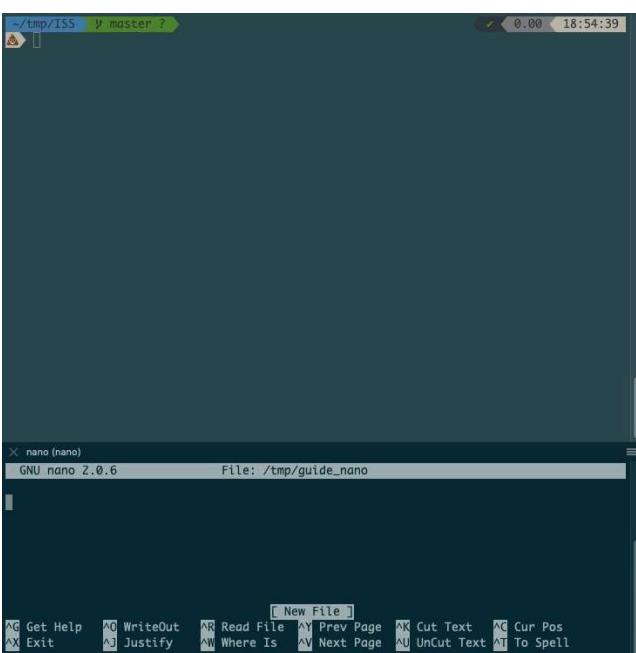
Une fois votre TP et votre rapport terminés, vous devez rendre votre travail (c'est à dire votre dossier avec tous les fichiers, ainsi que le README) sous la forme d'une archive tar compressée, autrement dit d'un tar.gz ou tgz. Pour ce faire, vous pouvez utiliser la commande [tar](#) de la façon suivante:

```
~/ISS/ $ tar cvf <nom_de_l_archive>.tgz <dossier_source>
```

Exemple pour le TP-01 :

```
~/ISS/ $ tar cvf tp_01.tgz tp_01
```

Pour finir, voici un screencast (plein de pouces bleus à Ilyas) montrant le rendu d'un TP :



En préambule, nous tenons à vous rappeler que, comme indiqué pendant le premier cours, il est indispensable pour vous d'avoir accès à un système Linux sur votre machine personnelle. Cela peut passer par un double boot (méthode conseillée) ou par l'utilisation d'une machine virtuelle sur un autre système. Les techniques présentées ici ne remplaceront pas un vrai système.

L'ensemble des ordinateurs de PPTI offre un service de connexion à distance au travers du protocole sécurisé [SSH](#). Ce protocole permet de vous authentifier sur la machine et d'ouvrir un terminal virtuel sur cette machine. Une fois connecté vous allez donc avoir un shell qui s'exécute sur la machine distante comme si vous étiez devant. Vous aurez donc accès à tout l'environnement des TP ainsi qu'aux fichiers de votre compte.

En pratique, il vous faudra vous connecter sur la passerelle, puis dans un deuxième temps sur la machine. (les machines des salles ne sont pas accessibles directement de l'extérieur), puis vers la machine ciblée. Pour vous connecter vous aurez besoin, d'un client (logiciel implémentant le protocole utilisé par un serveur). Il en existe pour les différents systèmes :

- Linux / MacOS : commande [ssh](#) (généralement dans le paquet [openssh](#))
- Windows : [putty](#) (attention de nombreux sites vous le proposent, mais pour la sécurité de votre système préférez toujours le site original - voir lien)
- Android : [connectbot](#)

Dans l'exemple suivant, on considère l'étudiant 1234567 qui souhaite se connecter sur la machine 3 de la salle 402 de la ppti. Il doit donc se connecter successivement à la passerelle, puis sur la machine (notez que dans la deuxième connexion il est inutile d'indiquer le login puisqu'il reste le même) :

```
moi@monOrdi ~ $ ssh 1234567@ssh.ufr-info-p6.jussieu.fr  
1234567@ssh.ufr-info-p6.jussieu.fr ~ $ ssh ppti-14-403-06  
1234567@ppti-14-402-03.ufr-info-p6.jussieu.fr ~ $ exit  
1234567@ssh.ufr-info-p6.jussieu.fr ~ $ ssh ppti-14-406-07  
1234567@ppti-14-406-07.ufr-info-p6.jussieu.fr ~ $ exit  
1234567@ssh.ufr-info-p6.jussieu.fr ~ $ exit  
moi@monOrdi ~ $
```

Pour simplifier votre connexion, vous pouvez automatiser le rebond sur la passerelle, et en profiter pour définir un alias, en ajoutant (crée-le s'il n'existe pas) dans le fichier [~/.ssh/config](#) de votre *home* les lignes suivantes en laissant les ??? . Il vous suffira alors de saisir directement par exemple **ssh ppti-14-403-06** ou **ssh ppti-14-407-06** :

```
Host ssh-ppti  
  HostName ssh.ufr-info-p6.jussieu.fr  
  User 1234567  
  
Host ppti-14-???-??  
  User 1234567  
  ProxyCommand ssh ssh-ppti -W %h:%p
```

Vous savez maintenant comment obtenir un shell sur une machine distante. Mais puisque tous se passe comme si vous étiez sur place, vos répertoires et fichiers locaux ne sont plus accessibles. Pour pouvoir envoyer et recevoir des fichiers vous allez devoir utiliser la commande [scp](#) qui utilise le protocole ssh pour échanger des fichiers. La syntaxe est la même qu'un *cp* classique. Mais les fichiers distants doivent être précédés par **login@addr:** . Vos fichiers étant accessibles sur la passerelle, inutile ici de rebondir vers une machine des salles de la ppti.

```
# Envoie le répertoire tp-01 se trouvant sur mon ordi à la ppti  
moi@monOrdi ~ $ scp -r tp-01/ 1234567@ssh.ufr-info-p6.jussieu.fr:  
# Récupère le fichier README du tp2 à la ppti vers mon répertoire courant  
moi@monOrdi ~ $ scp 1234567@ssh.ufr-info-p6.jussieu.fr:tp-02/README .
```



Comme la commande scp utilise ssh, vous pouvez utiliser les alias définis dans votre fichier de configuration. Avec les lignes données ci-dessus, il faudrait faire :

```
moi@monOrdi ~ $ scp -r tp-01/ ssh-ppti:
```

```
moi@monOrdi ~ $ scp ssh-ppti:README .
```

### Configuration d'un MacOS pour pouvoir faire les TP

Le bash installé avec les MacOS est une version datant de 2007 (la version 3.2) qui n'est pas compatible avec les sujets de TP de l'UE. Dans le document ci-dessous (merci Ilyas), vous apprendrez à installer la version 5.0 à partir du dépôt Homebrew. Attention cette installation n'est pas suffisante puisque cette nouvelle version ne remplace pas l'ancienne. Il vous faudra, comme indiqué dans le document, configurer votre système pour qu'il l'utilise par défaut.



[Tuto sur l'installation et la configuration de Bash 5 sur MacOS](#)

## Bibliographie

### Manuels

**Titre :** Unix, Linux Et Les Systèmes D'exploitation - Cours Et Exercices Corrigé

**Auteur :** Michel Divay

**Éditeur :** Dunod

**Avis :** Pour ceux qui cherchent un manuel. Ce livre couvre une très grande partie du programme des UE de système en L2 et en L3. Vous pouvez le lire en entier tout est accessible pour des étudiants en licence.

### Livres de référence en français :

**Titre :** *Systèmes d'exploitation, 3<sup>e</sup> (Troisième) édition (2003)*

**Auteur :** Andrew Tanenbaum

**Éditeur :** PEARSON Education

**Avis :** Couvre bien tous les concepts des systèmes d'exploitation. Passionnant, mais attention à ne pas vous y perdre. Attention, cette traduction n'a pas été mis-à-jour.

### Livres de référence en anglais :

**Titre :** *Modern Operating Systems, 4th (Fourth) Edition (2015)*

**Auteur :** Andrew Tanenbaum

**Éditeur :** PEARSON Education

**Avis :** Connue sous le nom de MOS, c'est la version originale. Elle a été corrigée et un peu étendue.

**Titre :** *Operating System Concepts, 10th Edition (2019)*

**Auteur :** Avi Silberschatz, Peter Galvin and Greg Gagne

**Éditeur :** Wiley

**Avis :** Très complet, il aborde à la fois les concepts techniques mais aussi les algorithmes mis-en-œuvre dans les systèmes d'exploitation. Un grand classique.

## Schémas du cours



[C'est quoi le système d'exploitation ?](#)



[Fork / Exec](#)



[Les waits](#)

## Semaine 01 - Introduction au Bash

Le but de cette première semaine est de comprendre le fonctionnement de l'interprète de commandes *bash* et d'apprendre à réaliser un petit script.

[TD 01](#)[TP 01](#)[Sources pour l'exercice 3 du TP-01](#)[Sources pour l'exercice 4 du TP-01](#)

---

## Semaine 02 - Introduction à la notion de processus

Le but de cette deuxième semaine est d'appréhender la notion de processus, d'étudier le lancement d'un programme depuis un terminal, d'apprendre à travailler avec la valeur renvoyée par le programme et de comprendre qu'une commande *Bash* n'est rien d'autre qu'un programme qu'on lance.

[TD 02](#)[TP 02](#)

---

## Semaine 03 - Les sources d'entrée pour les données d'un script

Le but de cette troisième semaine est d'approfondir l'étude des arborescences de processus commencée la semaine précédente. Il s'agit aussi de faire le tour des différentes sources d'entrée pour les données d'un script : paramètres, variables d'environnement, redirections du flux d'entrée standard et substitutions de commandes.

[TD 03](#)[TP 03](#)[Sources pour le TP-03](#)[Sources pour le TP-03 : chunks](#)

---

## Semaine 04 - Combinaison de commandes avec les tubes

Le but de cette quatrième semaine est d'approfondir l'étude des arborescences de processus, d'introduire la notion de variables d'environnement. Nous étudierons aussi les différents types de redirections et reviendrons sur la substitution de commandes introduite la semaine précédente.

[TD 04](#)[TP 04](#)[Sources pour le TP-04](#)

---

## Semanie 05 - Les expressions régulières

Le but de cette cinquième semaine est d'étudier les expressions régulières, tant d'un point de vue syntaxique, que pratique. Ainsi, nous verrons leur mise-en-œuvre dans le cadre de la commande *grep*.



[TD 05](#)[Dicos pour TD-05 \(deux versions\)](#)[TP 05](#)

---

## Semaine 06 - Édition automatisée

Le but de cette sixième semaine est d'étudier un exemple de commande permettant d'automatiser l'édition d'un flux. Comme le *grep*, ces commandes peuvent utiliser les expressions régulières étendues. Mais celles-ci s'en servent aussi pour formaliser des modifications à appliquer sur le contenu d'un flux de caractères. Les plus connus de ces commandes sont *awk* et *sed*. Nous avons ici choisi d'étudier cette dernière.

[TD 06](#)[TP 06](#)

---

## Semaine 07 - Cycle de vie d'un processus et exécution concurrente

Le but de cette septième semaine est d'étudier la commutation de processus, ainsi que l'automate décrivant la vie d'un processus. On y apprendra à lancer des commandes en parallèle, en s'attachant à observer les commutations de processus. Enfin, elle sera l'occasion d'introduire les concepts d'ordonnancement batch et temps partagé.

[TD 07](#)[TP 07](#)[Sources pour le TP-07](#)

---

## Semaine 08 - Parallélisation et synchronisation avec wait

Le but de cette huitième semaine est d'étudier la parallélisation de commandes et la synchronisation à l'aide de la commande *wait*. Nous y introduirons aussi la notion d'adoption et de processus zombie.

[TD 08](#)[TP 08](#)[Sources pour l'exercice 1 du TP-08](#)[Sources pour l'exercice 2 du TP-08](#)

---

## Semaine 09 - Les signaux

Le but de cette neuvième semaine est d'étudier la communication par signaux. On y étudiera l'envoi de signaux, mais aussi la redéfinition des handlers de signaux.

[TD 09](#)

## Semaine 10 - Concurrence, incohérence et synchronisation

Le but de cette dixième semaine est de mettre évidence que la commutation entre processus peut être à l'origine d'incohérence sur des ressources partagées. Nous y verrons aussi comment répondre à ce problème en utilisant un mécanisme de verrouillage.

[TD 10](#)[TP 10](#)[Sources pour le TP-10](#)

## Semaine 11 - L'autre synchronisation : la barrière

Cette onzième semaine revient sur les mécanismes de synchronisation étudiés la semaine dernière. Ainsi, nous avons déjà vu comment ces derniers permettent à plusieurs processus de travailler sur une même ressource partagée. Dans cette séance, nous verrons comment ces mêmes mécanismes permettent de mettre en place des barrières, i.e., d'attendre la fin d'une action ou d'un calcul fait par un autre processus.

[TD 11](#)[TP 11](#)

## Consignes pour le partielle, l'examen et la session de deuxième chance

Dans cette UE, tous les examens écrits se déroulent sans documents et portent, sauf indications contraires, sur tout ce qui a été vu jusqu'à l'épreuve. Cela comprend les TD et les TP, mais aussi les cours. Ainsi un tiers des épreuves porte sur des schémas et des connaissances vus en cours.

En absence de document, vous devez avoir un minimum de connaissance du Bash pour passer les épreuves. Outre, le fonctionnement des variables en bash, les variables spéciales (`$@`, `$#`, `$$`, `$!` et `$?`), le mécanisme de substitutions (et notamment *les métacaractères*, `$(..)` et `$((...))`), les structures de contrôle (**if**, **for** et **while**), les redirections et les tubes, vous devez connaître le fonctionnement des commandes suivantes :

- **echo** et son option `-n`
- **cat**
- **cd**
- **mkdir**
- **rm** et son option `-r` pour la suppression des répertoires
- **export** et **env**
- **read** et notamment sa forme **while read x y z ; do .... ; done < fichier** que l'on peut utiliser avec la variable **IFS** pour redéfinir le séparateur
- **test** (et sa forme `[ ]`) avec les opérateurs sur les fichiers (`-f`, `-d`, `-r` et `-w`), sur les valeurs entières (`-eq`, `-lt`, `-le`, `-gt` et `-ge`) et les ou/et (`-o` et `-a`)
- **head** et **tail** et leur option `-n` (avec valeur de l'option *positive* ou *négative*)
- **wc** et son option `-l`
- **tr** et son option `-s`
- **sort** et son option `-n` pour les valeurs numériques
- **uniq**
- **cut** et ses options `-d` et `-f`
- **grep**, son option `-v` pour inverser la sélection, les expressions régulières (activées avec `-E`) basées sur `: . [...] ? * +` et les références arrière (...), `\x`

- **sed**, son option **-n** et ses commandes **s/.../...**, **d** et **p**
- le lancement d'une commande avec **&** et **wait**
- **kill** et **trap**
- **exit**

Si besoin, les énoncés des exercices préciseront le fonctionnement d'autres commandes ou d'autres options.

## Annales

 [Partiel - Novembre 2019](#)

 [Partiel - Novembre 2019 - Correction](#)

 [Examen - Janvier 2020](#)

 [Examen - Janvier 2020 - Correction](#)

 [Partiel - Décembre 2020](#)

 [Partiel - Décembre 2020 - Correction](#)

 [Examen - Janvier 2021](#)

 [Examen - Janvier 2021 - Correction](#)

 [Partiel - Novembre 2021](#)

 [Partiel - Novembre 2021 - Correction](#)

 [Examen - Janvier 2022](#)

 [Examen - Janvier 2022 - Correction](#)

 [Partiel - Décembre 2022](#)

 [Partiel - Décembre 2022 - Correction](#)

Connecté sous le nom « Lili Zheng » (Déconnexion)

Accueil

Obtenir l'app mobile Condition générale d'utilisation et conservation des données

