



# **Systèmes UNIX**

---

## **Travaux Pratiques** **Systèmes d'exploitation et réseau** **TP**



## Consignes

### Durée :

- 14 heures

### Organisation :

- A réaliser individuellement

### Matériel :

- 1 postes de travail ;
- Le poste de travail dispose de :
  - 1 Interface réseau branchée sur le réseau de la salle.
- 1 machine virtuelle avec une distribution linux Debian :
  - 1 Interface réseau en connexion de pont sur l'interface réseau connectée au réseau de la salle ;

### Objectif du TP :

- Apprendre à configurer et exploiter les possibilités d'un système d'exploitation issu de UNIX.

### Etudiant :

- Prénom :

\_\_\_\_\_

- Nom :

\_\_\_\_\_



## « Système d'exploitation et réseau »

# I Le système de fichiers

## I.1 Utiliser le système de fichiers

1. Ouvrez un terminal en tant qu'utilisateur (pas en tant qu'administrateur) et affichez :
  - Votre nom de connexion.  

---
  - Votre `UID` (IDentifiant Utilisateur) et `GID` (IDentifiant de Groupe).  

---
  - Le nom de la machine de connexion.  

---
  - La liste des utilisateurs connectés à la même machine que vous.  

---
  - Le nom du répertoire courant.  

---
  - La liste de tous les fichiers du répertoire courant (fichiers cachés compris).  

---
2. Positionnez-vous sous le sous-répertoire de nom `local` du répertoire `/usr`  

---
3. Positionnez-vous sous votre répertoire de connexion.  

---
4. Affichez la liste de tous les fichiers de chacun des répertoires `/usr/local/bin`, `/bin` et `/usr/share`, en utilisant la désignation absolue ou relative.  

---



5. Créez, sous votre répertoire de connexion utilisateur, deux répertoires avec, respectivement, pour noms `tp` et `TMP`  

---
6. Copiez les fichiers `passwd`, `group` et `hosts` du répertoire `/etc` sous votre répertoire de connexion.  

---
7. Dupliquez `passwd`, sous votre répertoire de connexion, sous le nom `password`  

---
8. Déplacez le fichier `hosts` sous le répertoire `tp` en le renommant en `hôtes`  

---
9. Supprimez le fichier `password`  

---
10. Essayez de détruire le répertoire `tp`. Justifiez le message d'erreur obtenu.  

---

---
11. Déplacez `group` et `notes` vers `TMP`  

---
12. Détruisez le répertoire `tp`  

---
13. Renommez le répertoire `TMP` en `tmp`  

---



14. Affichez le contenu des fichiers `notes` et `passwd` page par page.

---

15. Observez la différence d'affichage de `ls -l tmp` et de `ls -ld tmp`

---

16. Donnez la date de dernière modification du répertoire `tmp`

---

17. Donnez la taille du fichier `notes`

---

## I.2 Utilitaires pour travailler sur les fichiers

### I.2.a Utilitaires `wget` et `tar`

1. Créez un répertoire `UNIX` qui servira pour toute cette partie et placez-vous sous ce répertoire.

---

2. Téléchargez l'archive `0.1.tar.gz` disponible à l'adresse suivante : <https://github.com/VirtuoWorks/tp-systemes-unix/archive/0.1.tar.gz> à l'aide de l'utilitaire `wget` dans votre répertoire courant.

---

3. Restaurez le contenu de l'archive `0.1.tar.gz`

---

4. Vérifiez le contenu de votre répertoire `UNIX` à l'aide de la commande `ls -l`



5. Affichez le contenu de ces fichiers.

**NB :** chaque ligne du fichier `fruits` comprend quatre champs : le stock, la ville, le nom du fruit et le prix. Chaque ligne du fichier `auto` comprend six champs séparés par des deux-points (":") : la marque, le modèle, l'année, le kilométrage, le prix et la couleur.

---

6. Supprimez l'archive `0.1.tar.gz`

---

### 1.2.b Comment utiliser `grep`, `cut` et `sort`

- Comment Afficher la liste des véhicules `gris` du fichier `auto`

➤ `grep gris auto`

*Ici `grep` recherche dans le fichier `auto` le mot `gris`*

➤ Etudiez les pages du manuel en ligne de `grep` pour connaître les autres options de recherche sur les fichiers sélectionnés. Attention : l'option `-r` (pour récursif) n'est pas disponible sur toutes les versions du manuel. Par ailleurs, sous GNU/Linux, par erreur, les pages du manuel en ligne en français de la commande `grep` ne mentionnent pas forcément l'option `-r` qui permet de sélectionner un ensemble de fichiers dans une arborescence plutôt que dans un seul répertoire.

- Comment afficher de la liste des modèles (2ème champ) du fichier `auto`

➤ `cut -d?:? -f2 auto`

*Ici `cut` recherche dans le fichier `auto` et sélectionne pour l'affichage, avec l'option `-f`, les mots figurant comme 2ème champs. Le séparateur de champ n'est pas un espace ou une tabulation, il faut donc le spécifier avec l'option `-d`.*

➤ Etudiez les pages du manuel en ligne de `cut`.

- Comment trier des autos en fonction du modèle (2ème champ)

➤ `sort -t?:? -k2 auto`



*Le séparateur de champ n'est pas un espace ou une tabulation, il faut donc le spécifier avec l'option `-t`.*

- Etudiez les pages du manuel en ligne de `sort` pour les autres options de tri.

### I.2.c Rechercher avec `grep` et `cut`

1. Affichez la liste des véhicules de l'année 89 du fichier `auto`.  

---
2. Affichez la liste des véhicules du fichier `auto` qui ne sont pas rouges.  

---
3. Affichez toutes les notes de Gregory écrites dans le fichier `notes`.

**Remarque :** pour ignorer les différences majuscules/minuscules dans le motif recherché, utilisez une option de `grep`.

---

4. Affichez toutes les couleurs de voiture présentes dans le fichier `auto`.  

---

### I.2.d Trier avec `sort`

1. Triez le fichier `auto`  

---
2. Triez le fichier `fruits` avec et sans l'option numérique. Que constatez-vous ?  

---



3. Triez le fichier `fruits` sur le troisième champ (nom du fruit) et, en cas d'égalité, sur le deuxième champ (nom de la ville). Comment sont classées les lignes pour lesquelles ces deux champs sont identiques ?

---

---

### I.2.e Finalement

1. Allez dans le répertoire père du répertoire `UNIX`.
2. Créez une archive que vous nommerez `mon_tp.tar.gz` et contenant l'ensemble des fichiers du répertoire `UNIX`.

---

## I.3 Comprendre les droits d'accès

1. Placez-vous dans votre répertoire de connexion utilisateur. Vérifier que le répertoire `tmp` créé au début du TP est bien présent.
2. Positionnez les droits d'accès `rw- r-x ---` pour `tmp`
3. Positionnez les droits d'accès `rw- r-- ---` pour le fichier `notes`. Lisez le contenu de `notes`

---

---





**Remarque :** la lecture du fichier `tmp/hotes` est permise. Le fichier peut être néanmoins vide.

---

4. Retirez pour le propriétaire le droit en lecture sur le fichier `hotes`. Essayez de lire le contenu de `hotes`. Justifiez le message d'erreur obtenu.
- 

5. Remettez pour le propriétaire le droit en lecture sur le fichier `hotes`
- 

6. Retirez pour le propriétaire le droit en écriture sur le répertoire `tmp`. Essayez de détruire `hotes`. Justifiez le message d'erreur obtenu.
- 

7. Retirez pour le propriétaire le droit en lecture sur le répertoire `tmp`. Essayez de lister le contenu de `tmp`. Justifiez le message d'erreur obtenu.
- 

8. Lisez le contenu de `hotes`. Expliquez pourquoi vous pouvez le lire.
-



9. Retirez pour le propriétaire le droit `x` sur le répertoire `tmp`. Essayez de vous positionner sur ce répertoire. Justifiez le message d'erreur obtenu.

---

---

10. Essayez de lire le contenu de `notes`. Justifiez le message d'erreur obtenu.

---

---

11. Positionnez pour le propriétaire les droits d'accès `rwX` sur le répertoire `tmp`

---

---

12. Essayez de vous donner le droit en écriture sur le répertoire de connexion utilisateur d'un membre de votre groupe utilisateur.

---

---

13. Justifiez le message d'erreur obtenu. Supprimez le fichier `hôtes`

---

---

14. Supprimez le répertoire `tmp`

---

---

## I.4 Comprendre les liens symboliques et physiques

### Préparation



1. Créez l'arborescence de la Figure 1 dans votre répertoire UNIX

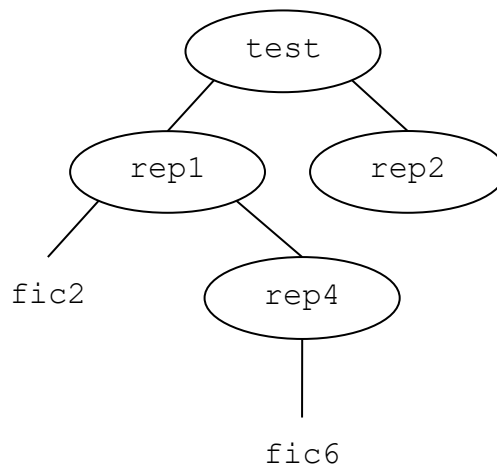


Figure 1 : Arborescence initiale

2. Donnez trois façons de désigner le fichier `fic6` depuis votre répertoire de connexion utilisateur.

---

---

---

3. À l'aide de la commande `echo`, écrivez *Il fait beau aujourd'hui*. Dans le fichier `fic6`

---

4. À l'aide de la commande `cat`, affichez le contenu du fichier `fic6` depuis le répertoire `rep2`

---

5. L'option `-l` de la commande `ls` permet entre autres d'observer les droits d'un fichier et/ou répertoire. À quoi correspondent les autres informations que l'on obtient grâce à cette commande ?

---



## Création de liens physiques et symboliques sur des fichiers

6. À l'aide de la commande `ln`, créez un lien physique (ou matériel) sur le fichier `fic6` dans `test` sous le nom `lpfic6`.  

---
7. Modifiez le fichier `lpfic6` à l'aide de la commande `echo`. Que constatez-vous dans le fichier `fic6` ? Réciproquement, modifiez `fic6` et lisez `lpfic6`. Que concluez-vous ?  

---

---
8. Modifiez les droits d'accès au fichier `fic6` pour les membres du groupe utilisateur. Que constatez-vous pour `lpfic6` ? Donnez une explication.  

---

---
9. La commande `ln` peut aussi créer des liens symboliques avec l'option `-s`. Créez un lien symbolique sur le fichier `fic6` dans `test` que vous appellerez `lsfic6`  

---



10. Regardez toutes les informations concernant les fichiers `lpfic6` et `lsfic6`. Quelles différences notez-vous ?

---

11. Essayez de modifier les droits d'accès au fichier `lsfic6`. Que constatez-vous ?

---

12. Modifiez les droits d'accès au répertoire `rep1` pour ne plus y avoir accès. Essayez d'afficher le contenu de `lpfic6` et `lsfic6`. Que constatez-vous ? Donnez une explication.

---

---

13. Modifiez de nouveau les droits d'accès au répertoire `rep1` pour y avoir de nouveau accès. Déplacez le fichier `fic6` dans le répertoire `rep1`. Essayez d'afficher le contenu de `lpfic6` et `lsfic6`. Que constatez-vous ?

---

---

14. Redéplacez le fichier `fic6` dans le répertoire `rep4`. Essayez à nouveau d'afficher le contenu de `lpfic6` et `lsfic6`. Supprimez le fichier `fic6` puis recommencez. Que constatez-vous ?

---

---



---

## Création de liens physiques et symboliques sur des répertoires

15. Recréez le fichier `fic6` dans le répertoire `rep4`.
- 
16. Observez le répertoire `rep2`. Combien y a-t-il de liens sur ce répertoires ? À quoi correspondent-ils ?
- 
17. Dans le répertoire `rep2`, créez un sous-répertoire `rep3`. Combien y a-t-il maintenant de liens sur le répertoire `rep2` ? Expliquez ?
- 
18. Créez un lien physique `lprep4` du répertoire `rep4` dans le répertoire `test`. Que remarquez-vous ?
- 

Avec les implémentations existant actuellement, seul le super-utilisateur peut créer un lien matériel sur un répertoire, et encore, ce n'est pas toujours possible. Cependant, on peut toujours créer un lien symbolique sur un répertoire.



19. Créez un lien symbolique `lsrep4` du répertoire `rep4` dans le répertoire `test`
20. Créez un lien symbolique `lslsrep4` du lien `lsrep4` dans le répertoire `rep2`
21. Donnez trois manières différentes de se déplacer dans le répertoire `rep4` à partir du répertoire `test`
22. Que se passe-t-il si on utilise la commande `ls -Ral` sur `lsrep4` et sur `lsrep4/` ?

À la fin de cette partie, vous devriez avoir l'arborescence de la Figure 2 ci-après.

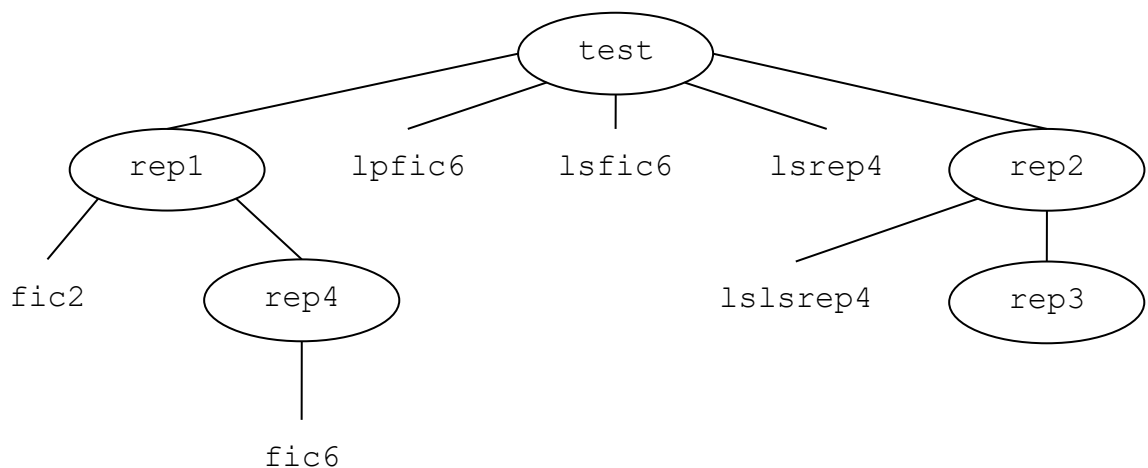


Figure 1 : Arborescence finale



## I.5 Mise en évidence des i-nœuds (inodes)

1. À l'aide de la commande `ls` munie de l'option appropriée, observez le numéro d'i-nœud du fichier `fic2`. Copiez le fichier `fic2` dans le répertoire `rep3`. Quel est son numéro d'i-nœud ?  

---
2. Changez le nom de ce dernier fichier, pour l'appeler `fic6`. Le numéro d'i-nœud change-t-il ?  

---
3. Comparez les numéros d'i-nœud entre le fichier `fic6` du répertoire `rep4`, `lpfic6` et `lsfic6`. Que remarquez-vous ? Expliquez plus clairement ce que vous aviez observé dans la partie « Création de liens physiques et symboliques sur des fichiers ».  

---

## II Les redirections et les tubes

### II.1 Les redirections

#### II.1.a Les bases

1. Quel est le contenu du fichier `temp` après chacune des étapes suivantes :
  - `cat /etc/group > temp`  

---





- `cat /etc/passwd > temp`

---

- `cat /etc/group >> temp`

---

2. Quelle est la différence entre les deux commandes `wc temp` et `wc < temp` ?

---

3. Quelle est la différence entre les deux commandes `ls ; who > temp` et `(ls ; who) > temp` ?

---

4. Supprimer le fichier `temp`

---

## II.1.b Le fichier spécial `/dev/null`

1. Afficher le contenu du fichier `/dev/null`

---

2. Afficher la taille du fichier `/dev/null`

---

3. Ajouter le contenu du fichier `/etc/passwd` à la fin du fichier `/dev/null`

---

4. Afficher le contenu du fichier `/dev/null`

---



5. Qu'en concluez-vous ?

---

6. Quelle spécificité du fichier `/dev/null` permet un tel comportement ?

---

### II.1.c Exercices avec les redirections

- Modifier la commande `find / -print` afin que :
  - Les erreurs soient redirigées vers le fichier spécial `/dev/null`
- Les données et les erreurs produites par la commande soient redirigées vers le fichier `/dev/null`
- A partir du terminal, créer un fichier `pi.txt` dont le contenu est :

Que j'aime à faire apprendre un nombre utile aux sages !  
Glorieux Archimède, artiste, ingénieur,  
Toi de qui Syracuse aime encore la gloire,  
Soit ton nom conservé par de savants grimoires !  
Jadis, mystérieux, un problème bloquait  
Tout l'admirable procédé, l'oeuvre grandiose  
Que Pythagore découvrit aux anciens Grecs.  
O quadrature ! Vieux tourment du philosophe  
Insoluble rondeur, trop longtemps vous avez  
Défié Pythagore et ses imitateurs.  
Comment intégrer l'espace plan circulaire ?  
Former un triangle auquel il équivaudra ?  
Nouvelle invention : Archimède inscrira  
Dedans un hexagone ; appréciera son aire  
Fonction du rayon. Pas trop ne s'y tiendra :  
Dédoublera chaque élément antérieur ;  
Toujours de l'orbe calculée approchera ;  
Définira limite ; enfin, l'arc, le limiteur  
De cet inquiétant cercle, ennemi trop rebelle  
Professeur, enseignez son problème avec zèle

*(Ne re-saisissez pas la totalité du texte dans votre réponse)*

---



- A partir du fichier `/etc/passwd`, afficher le nom de l'utilisateur dont le login contient `gdm`. Pour cela :
  - Sélectionner les lignes du fichier `/etc/passwd` contenant la chaîne de caractères `gdm` et stocker le résultat dans le fichier `temp`
  - Sélectionner le cinquième champ (ce champ s'appelle champ `GECOS`, il contient des informations générales au sujet du compte utilisateur) de chaque ligne du fichier `temp`
  - Supprimer le fichier `temp`
- 

## II.2 Les tubes

1. A partir du fichier `/etc/passwd`, afficher le nom de l'utilisateur dont le login contient `gdm`, mais sans utiliser de fichier temporaire.
  2. Afficher les lignes 3 à 7 du fichier `/etc/passwd`
  3. A partir du contenu du fichier `/etc/passwd`, déterminer combien d'utilisateurs utilisent l'interpréteur de commandes `bash`
  4. Combien y a-t-il de commandes commençant par le mot `less` dans le répertoire `/bin` ?
  5. La commande `ps -aux` affiche la liste des processus sur la machine
    - Afficher le nombre de processus associés à l'utilisateur `root`
-



- Afficher la liste des utilisateurs ayant au moins un processus s'exécutant sur la machine
- Afficher dans l'ordre décroissant le nombre de processus s'exécutant sur la machine pour chacun des utilisateurs

## III Les Processus

### III.1 Manipuler un processus

1. Taper la commande `xeyes` dans un terminal. Que se passe-t-il ? Le terminal est-il toujours utilisable ?
2. Taper la combinaison de touches `CTRL+C` dans le terminal bloqué. Que s'est-il passé ? Quel est l'état du terminal ?
3. Taper à nouveau la commande `xeyes` dans un terminal. Une fois la commande lancée, taper la combinaison de touches `CTRL+Z`. Quel est le comportement du terminal ? Quel est le comportement de `xeyes` ?
4. Sans rien toucher au terminal utilisé pour la question 3, ouvrir un nouveau terminal et lister les processus actifs sur la machine (`ps -aux`). Quel est l'état



du processus `xeyes` ? Si vous avez du mal à répondre, refaire les manipulations des questions 3 et 4 à la suite.

---

5. Dans le terminal d'où la commande `xeyes` a été lancée et qui a été libéré par la combinaison de touches `CTRL+Z`, taper maintenant la commande `bg`. Que se passe-t-il au niveau du terminal ? Que se passe-t-il au niveau des yeux ?
- 

6. Dans un terminal autre que celui où la commande `xeyes` a été lancée, lister les processus actifs sur la machine et chercher le processus `xeyes`. Dans quel état se trouve-t-il ?
- 

7. Tenter d'arrêter la commande `xeyes` depuis le terminal d'où elle a été lancée en tapant la combinaison de touches `CTRL+C`. Quel est le résultat ?
- 

8. Depuis le terminal où `xeyes` a été lancé puis détaché, taper la commande `fg`. Le terminal est-il toujours utilisable ?
- 

9. Arrêter la commande `xeyes` depuis le terminal d'où elle a été lancée en tapant la combinaison de touches `CTRL+C`.

10. Vérifier que le processus `xeyes` a bien disparu en utilisant la commande `ps -aux`.



## III.2 Envoyer des signaux

Le contrôle de processus peut s'effectuer depuis le terminal d'où la commande correspondante a été lancée mais aussi depuis d'autres terminaux grâce à l'utilisation de signaux.

1. La commande `kill -l` liste les signaux disponibles sur notre système. Taper cette commande dans un terminal, combien de signaux sont listés ?  

---
2. Depuis un terminal, lancer la commande `xeyes`. Depuis un autre terminal, rechercher le PID de la commande `xeyes` et lui envoyer le signal 19 (SIGSTOP) grâce à la commande `kill -19 <pid de xeyes>`. Que se passe-t-il ?  

---
3. A l'aide des commandes `kill -l`, du `man` ou d'internet, trouver et envoyer un signal au processus `xeyes` afin qu'il reprenne son exécution.  

---
4. Depuis le terminal utilisé pour envoyer le précédent signal, envoyer le signal -9 (SIGKILL) au processus `xeyes`. Quel est le résultat ?  

---

## III.3 Du programme au processus

### III.3.a Analyser un code source

- Objectif : étude des fichiers de code source d'un jeu.
- Consigne : nous nous proposons de lister tous les types de fichiers présents dans ce répertoire. Classiquement, le type de fichier est caractérisé par l'extension du fichier (l'extension étant le(s) caractère(s) qui sui(ven)t le dernier point (« . ») dans le nom du fichier).



1. Télécharger à l'aide `wget` le code source au format `tgz` du jeu `Armagetron AD` à l'adresse : <http://www.armagetronad.org/downloads.php> (**rubrique Source**).  

---
2. Décompresser l'archive dans un sous-dossier de votre dossier utilisateur.  

---
3. Positionnez-vous dans le répertoire des sources (dossier `armagetronad-0.2.8.3.3/src`).  

---
4. Tapez la ligne de commande permettant de déterminer le nombre de fichiers contenus dans ce répertoire.  

---
5. Tapez la commande qui liste le contenu du répertoire, en ajoutant l'option permettant d'afficher une seule entrée par ligne.  

---
6. Complétez cette commande pour qu'elle ne conserve en sortie que les extensions des fichiers listés (pensez à utiliser la commande `cut`).  

---



7. Les fichiers `.cpp` contiennent le code « source » et les fichiers `.h` le code d'interface. À l'aide de la commande `ls`, listez les huit premiers fichiers d'extension `.h` ou `.cpp`

---
8. La commande précédente donne à penser que pour chaque fichier `.h`, il y a un fichier `.cpp`. Comptez le nombre de fichiers `.h`, puis le nombre de fichiers `.cpp`

---
9. Stockez dans le fichier `../listeH.txt` la liste des fichiers `.h` sans leur extension (par exemple si il existe un fichier `./src/about.h`, votre fichier `../listeH.txt` doit contenir la ligne `about`).

---
10. Idem avec `../listeC.txt` pour la liste des fichiers `.cpp` sans leur extension.

---

### III.3.b Construire un programme

- Objectif : compiler les sources et faire l'édition des liens pour obtenir l'exécutable.
  - Consigne : suivez bien les instructions les unes après les autres pour ne pas vous perdre.
1. Déplacez-vous dans le répertoire `armagetronad-0.2.8.3.3`
  2. Tapez la commande `./configure` pour configurer l'environnement logiciel de Armagetron AD. Pour que, par la suite, les binaires d'Armagetron AD soient créés dans le `\dossier\de\votre\choix` vous devez exécuter `./configure --`





`prefix=\dossier\de\votre\choix` (l'option `--prefix` permet d'indiquer le dossier dans lequel `make` devra créer les binaires). Bien entendu le dossier doit avoir été créé au préalable. Par ailleurs, si vous avez besoin d'installer la librairie `libxml2`, vous pouvez télécharger la librairie sur cette [page](#) ([libxml2-2.9.10.tar.gz](#)) et l'installer d'après les explications fournies [ici](#) ([voir le point 2 de la rubrique Developer Corner](#)).

3. Après un certain nombre de messages, vous obtenez le message `Type 'gmake' to build armagetronad-0.2.8.3.3` qui suggère de taper `gmake`. Cherchons donc à déterminer la différence entre les outils `gmake` et `make`. Commençons par déterminer le chemin absolu vers la commande `gmake`. Même question pour la commande `make`.
- 

4. Listez de manière longue, c'est-à-dire en affichant les attributs, les fichiers exécutables correspondant à ces deux commandes.
- 

5. Qu'en concluez-vous ?
- 

6. Comment pourrait-on répondre à la question 3 sans taper explicitement le chemin vers les commandes `gmake` et `make` ?
- 

7. Aidez-vous de [cette page de la documentation](#) pour compiler `Armagetron AD`. Expliquez ce que vous avez fait.
-



8. Lancez le jeu et faites une partie. Donnez la commande.

---

### III.3.c Jouer en réseau !

- Objectif : mettre en œuvre les fonctionnalités de jeu en réseau de Armagetron AD.
- Consigne : suivez attentivement les instructions.

Pour pouvoir jouer en réseau, vous avez besoin de connaître l'adresse "inet" (Internet en abrégé) de votre machine. La commande `ifconfig` vous fournit cette information. Toutefois, elle n'est pas dans votre `PATH`. Localisez cette commande (sans utiliser la commande `find`, car elle prendrait trop de temps). Attention ! La commande permettant de réaliser cette opération n'a pas été vue en cours. Utilisez les pages du manuel afin de trouver la bonne commande.

Maintenant que vous savez où se trouve cette commande, exécutez-la en filtrant sa sortie pour ne conserver que les lignes contenant `inet adr: .` Notez les quatre nombres séparés par des points qui suivent `inet adr:`

Ceci est votre adresse IP. Essayez de configurer Armagetron AD pour jouer en réseau avec une autre personne.

Expliquez votre démarche.

---



## IV Scripts UNIX

### IV.1.a Les variables

Exécuter les commandes suivantes et indiquer le résultat obtenu.

1. `i=4`

2. `echo $i`

---

3. `echo $i$i`

---

4. `echo $iime`

---

5. `echo ${i}ime`

---

6. `echo "${i}ime"`

---

7. `echo '${i}ime'`

---

8. `i=`expr $i + 1``

9. `echo $i`



---

10. Donner une autre façon d'incrémenter la variable `i`.

---

#### IV.1.b Scripts avec paramètres

1. Écrire un script affichant le nombre de processus en cours d'exécution sur la machine pour l'utilisateur dont le `logname` est fourni en paramètre.

---

2. Écrire un script affichant les `l` lignes suivant la première occurrence de l'expression régulière `e` dans le fichier `f`. Les valeurs `l`, `e` et `f` doivent être des paramètres du script. Note : l'utilisation des options `-A` et `-after-context` de la commande `grep` est interdit.

---



### IV.1.c Boucles et récursions

1. À l'aide d'une boucle `for`, afficher les chiffres de 0 à 9.
- 

2. Afficher les nombres jusqu'à 99, toujours sans utiliser une boucle `while`
-



3. Écrire un script affichant les nombres de  $n$  à  $p$  ( $n$  et  $p$  étant des paramètres du script). Note : pour simplifier, on choisira  $n$  et  $p$  strictement positifs.

---

4. Écrire une nouvelle version de ce script sans utiliser de boucle.

---



#### IV.1.d Script complexe

1. Écrire un script affichant la nature d'un fichier (régulier, répertoire, périphérique, lien symbolique, tube nommé ou socket).
- 

2. Compléter le script précédent de sorte que plusieurs fichiers puissent être précisés sur la ligne de commande.
-



3. Compléter le script précédent de sorte que, pour un répertoire, ce soit la nature des fichiers contenus dans le répertoire qui soit affiché et non la nature du répertoire lui-même. Gérer l'utilisation d'une option (par exemple `-d`) qui annule la modification précédente.
- 

4. Compléter le script précédent de sorte que, pour un lien, ce soit la nature du fichier cible qui soit affiché et non la nature du lien lui-même. Gérer l'utilisation d'une option (par exemple `-l`) qui annule la modification précédente.
-