

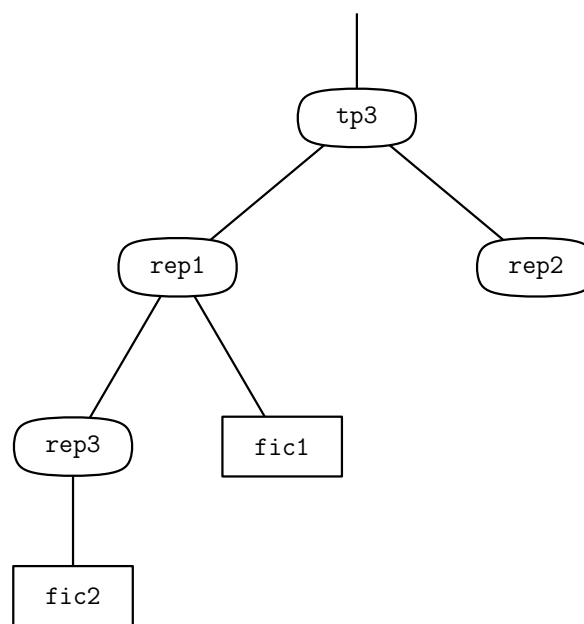
Introduction aux systèmes d'exploitation (IS1)

TP 3 – Liens, i-nœuds

Dans ce TP, nous allons étudier plus précisément la conception du système de fichiers UNIX.

Exercice 1 – arborescence de travail

1. Pour commencer, créez l'arborescence suivante dans votre répertoire `~/IS1` :

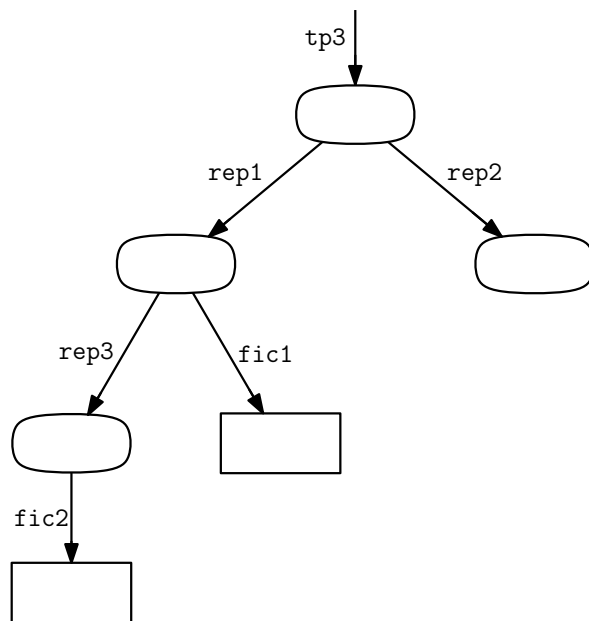


Le fichier `fic1` devra contenir le texte `Il fait beau aujourd'hui`, et `fic2` contiendra `Demain est un autre jour`.

2. À l'aide d'une des options de la commande `ls`, vérifiez que l'arborescence que vous avez créée est bien conforme au schéma.

Un système UNIX n'identifie pas un fichier par son nom, mais par une structure appelée *i-nœud* (*inode* en anglais pour *index node*), stockée dans une table et identifiée dans cette table par un numéro. Un i-nœud permet de stocker toutes les caractéristiques concernant le fichier correspondant : son type (fichier "régulier" ou répertoire), l'adresse du ou des emplacement(s) où se trouve son contenu, mais également les permissions accordées, la date de dernière modification...

Un répertoire est en fait une liste de couples formés d'une référence et d'un numéro. On dit qu'un fichier `fic` "se trouve" dans un répertoire `rep` si le répertoire `rep` contient le couple associant le nom `fic` au numéro d'i-nœud correspondant. On appelle cela un *lien* du nom `fic` vers l'i-nœud. Le schéma suivant est donc plus conforme à l'organisation réelle du système de fichiers.



Exercice 2 – les numéros d'i-nœud – retour sur « cp » et«

La commande « ln » peut aussi créer des liens *symboliques* avec l'option -s. Il s'agit de fichiers spéciaux contenant la référence d'un autre fichier, vers lequel pointe le lien. Contrairement aux liens physiques, il s'agit d'une relation dissymétrique.

Exercice 4 – liens symboliques vs liens physiques

1. Créez dans tp3 un lien physique l_p_fi c2 et un lien symbolique l_s_fi c2 vers le fichier fi c2. Comparez leurs contenus, puis leurs numéros d'i-nœud, ainsi que leurs autres caractéristiques.
2. Essayez de modifier les droits d'accès à l_s_fi c2. Que constatez-vous ?
3. Modifiez les droits d'accès au répertoire rep2 pour ne plus y avoir accès. Essayez d'afficher le contenu de l_p_fi c2 et l_s_fi c2. Que constatez-vous ? Pourquoi ? Rétablissez les droits.
4. Déplacez fi c2 dans rep1, puis essayez d'afficher le contenu de l_p_fi c2 et l_s_fi c2. Que constatez-vous ? Pourquoi ?
5. Créez un nouveau fichier fi c2 dans rep2 contenant le texte *Tel un phénix, je renais de mes cendres*. Affichez le contenu de l_p_fi c2 et l_s_fi c2. Expliquez.
6. Déplacez l_s_fi c2 dans rep1 puis tentez d'afficher son contenu. Expliquez.
7. Créez dans tp3 un lien symbolique l_s_rep3 vers rep3.
8. Créez dans rep2 un lien symbolique l_s_l_s_rep3 vers l_s_rep3.
9. Donnez trois manières différentes de se déplacer dans le répertoire rep3 à partir du répertoire tp3. En utilisant successivement ces trois méthodes, déplacez-vous dans le répertoire rep3 puis comparez les réponses de « pwd » et « pwd -P » ; remontez ensuite dans le répertoire parent à l'aide de la commande « cd .. ». Que remarquez-vous ?
10. Que se passe-t-il si on utilise la commande « ls -R » sur l_s_rep3 ? et sur l_s_l_s_rep3 ?
11. Comment arriver à la situation (gênante !) suivante ?

```
~$ ls -l
total 16
lrwxr-xr-x 1 titi staff 4 Sep 20 18:41 fi c1 -> fi c2
lrwxr-xr-x 1 titi staff 4 Sep 20 18:40 fi c2 -> fi c1
```