*Linux: appels système (I2181-B)*

*SOLUTION*

### 4. Exercice avec redirection de l’entrée standard

Reprenez l’exercice 2 et testez-le en redirigeant l’entrée standard :

lireEtEcrire Maj min < fichierIn

Le résultat est-il celui qui est attendu ? Pourquoi ?

**NON car dans un terminal, read lit dans le buffer système dès que la touche ENTER est enfoncée. Le terminal/clavier est un « character device » donc les caractères ‘\n’ sont bien traités comme des passages à la ligne.**

**Dans le cas où la lecture se fait dans un fichier (un fichier est un « block device »!), le caractère de fin de ligne '\n' est considéré comme n'importe quel caractère. read considère donc le fichier comme une longue chaîne et n'interrompra la lecture que si le nombre maximal de caractères demandé a été lu.**

**$ ./ex1.2 Maj min < lievreETtortue.txt**

**$ cat min**

et leur fait arpenter les landes.

Ayant, dis-je, du temps de reste pour brouter,

**$ cat Maj**

Moi l'emporter ! et que serait-ce

si vous portiez une maison ?

**Les 2 lignes sont écrites dans min car elles font exactement 81 caractères avec le dernier caractère = '\n'.**

**Les 2 dernières lignes du fichier sont copiées dans Maj car elles font moins de 81 caractères (car fin de fichier atteinte).**

**Ces lignes sont donc les seules qui ne sont pas ignorées par le programme.**

Comment faut-il modifier le programme pour qu’il produise le même effet, qu’il y ait redirection ou pas ?

**Pour traiter les lignes une par une, il faudrait traiter le buffer de lecture caractère par caractère : le programme détecte si le caractère qui a été lu est une fin de ligne '\n'.**

**🡪 cf. ex3.3.c**