

**OBHPC**  
**TD1**  
**Bash** sous Linux

**Entrées et sorties:**

- **0** standard input (**stdin**)
- **1** standard output (**stdout**)
- **2** standard error (**stderr**)

**Commandes:**

man  
echo  
touch  
ls  
mkdir  
rm  
mv  
cp  
cat  
less  
more  
wc  
tail  
head  
sed  
grep  
sleep  
kill  
killall  
read  
tar  
cut  
chmod  
chown  
gcc/make  
sort  
uniq

**Concepts:**

- Redirection des sorties vers des fichiers avec les symboles > et >>
- Redirection des sorties vers d'autres commandes avec les pipes (symbole | )
- Détachement des processus avec le symbole &
- Expressions régulières
- Scripting **bash** (conditions, boucles, ...)

## **Enonces:**

### **Section1**

L'objectif de ce TP est d'analyser un fichier **CSV** (Comma Separated Values) contenant la liste des instructions **x86** afin de les classer en plusieurs categories: instructions memoires, instructions arithmetiques et instructions de branchement en utilisant les outils/commandes disponibles dans un environnement **Linux**.

0 – Dans le repertoire **0** de **TD1**, creer a l'aide de la commande **mkdir**, les repertoires suivants:

**x86\_mem**, **x86\_arith**, **x86\_branch**.

1 – En utilisant la commande **grep**, isolez les instructions memoire dans un fichier **mem.csv**.

2 – Stocker le fichier **mem.csv** dans le repertoire **x86\_mem**.

3 – En utilisant la commande **grep**, isolez les instructions memoire dans un fichier **arith.csv**.

4 – Stocker le fichier **mem.csv** dans le repertoire **x86\_arith**.

5 – En utilisant la commande **grep**, isolez les instructions memoire dans un fichier **branch.csv**.

6 – Stocker le fichier **mem.csv** dans le repertoire **x86\_branch**.

7 – A l'aide de la commande **wc**, calculer le nombre d'instructions total ainsi que le nombre d'instructions memoires, arithmetiques et de branchement.

8 – Filtrer, a l'aide de **grep**, les instructions memoires du jeux d'instructions **SSE** dans le fichier **sse.csv**, les instructions **AVX** dans le fichier **avx.csv**, les instructions **AVX2** dans le fichier **avx2.csv** et pour finir, les instructions **AVX512** dans le fichier **avx512.csv**.

9 – Meme chose pour les instructions arithmetiques.

10 – Combien d'instructions possedent des operandes implicites? (analyser la colonne 14)

11 – Combien d'instructions gerent une taille de donnees de 64 bits?

12 – Automatiser tout le processus avec un script shell.

### **Section 2**

Ecrire un script shell qui compile et execute les programmes **p2** et **p3** situes dans le repertoire **1**, qui les execute avec les parametres de ligne de commande suivants: **10 10** et qui calcul la difference entre chacune des entrees des fichiers **c\_f32.dat** et **c\_f64.dat** generes par les deux programmes.

L'objectif etant de comparer la precision flottante simple et double.