

TP 4 – awk

1 Bash et Awk

- Pour vous familiariser avec *awk*, en utilisant l'exemple du cours

```
12 citrons fruit
10 ananas fruit
35 carottes legume
3  poires legume
```

- Créez un fichier `memo.txt`
- Réalisez dans un premier temps le (re)formatage des lignes (`p20`)
- Puis le traitement avant et après le traitement des lignes (`BEGIN` et `END`) (`p20`)

- Pour tester un autre séparateur de champ, affichez le contenu de la variable d'environnement `PATH` en colonne en utilisant l'option `-F` de *awk*. La commande commencera par `echo $PATH | awk` Vous serez amené à utiliser `for` et la variable `NF` (nombre de champs).

Exemple pour `PATH = /usr/bin:/usr/local/bin:/home/truc`, on devra obtenir :

```
/usr/bin
/usr/local/bin
/home/truc
```

- Dans un script *bash*, réalisez l'affichage des informations globales sur la mémoire du système :
 - affichez la taille d'une adresse virtuelle et d'une adresse physique (informations se trouvant dans le fichier `/proc/cpuinfo`),
 - ainsi que la taille totale de la mémoire, la taille de la mémoire libre et la taille des tables de pages (dans `/proc/meminfo`).

Vous utiliserez *awk* pour récupérer la bonne valeur dans la bonne ligne (le séparateur reste l'espace). On suppose que l'on connaît l'emplacement et la valeur de certains champs : on sait que "MemTotal :" doit être le premier champ (`$1`) de la ligne contenant la taille de la mémoire et on sait que sa valeur sera dans deuxième champ (`$2`).

- Dans la suite du script, on souhaite afficher le pourcentage de la mémoire que vous utilisez. Pour récupérer les lignes concernant vos processus en cours, vous utiliserez : `ps aux | grep "votrelogin"`¹ (pour connaître la signification des colonnes lancer d'abord `ps aux` seul). Nous n'utiliserons pas la colonne `%MEM` (car elle est arrondie donc la somme sera trop approximative). Vous utiliserez la colonne `RSS` qui indique la partie de la mémoire centrale utilisée par le processus. Pour extraire les données et faire les calculs, vous utiliserez *Awk*

1. la première colonne de cette commande doit contenir les noms des logins, il est possible que cela soit un numéro. Gardez `grep "votrelogin"`.

(pensez aux blocs BEGIN et END). Pour le %, utilisez la valeur de la mÃmoire totale obtenue à la question précédente (utilisez l’option -v pour utiliser une variable).

- ▶ Ensuite, dans le script, pour chaque ligne (de la commande **ps** précédente), récupérez le numéro du processus puis affichez la taille maximale de la pile (*stack*) utilisateur autorisée (à trouver dans le fichier `/proc/numero/limits`), ainsi que la taille de la mÃmoire virtuelle utilisée, de la pile utilisateur, des données, du texte et de la table de pages (respectivement : `VmSize`, `VmStk`, `VmData`, `VmExe`, `VmPTE` dans le fichier `/proc/numero/status`).

aide en *bash* :

- 🔗 On appelle *Soft limit*, la limite configurée et *Hard limit*, la limite maximum configurable ; vous prendrez la *Soft limit*.
- 🔗 Pour info : on peut retrouver ces limites avec la commande `ulimit -a` mais on ne l’utilisera pas dans ce TP ;-).
- 🔗 Pour info : vous trouverez des détails dans le fichier texte mis à votre disposition sur ARCHE sur le contenu du répertoire *proc*. Pour ce TP, on ne demande pas de le lire ;-). Attention, il s’agit de la version pour le noyau 2.2 donc des différences peuvent apparaître.

Remarque :

- 🔗 En début de script, n’oubliez pas d’ajouter des commentaires pour expliquer ce que fait le script et comment l’utiliser (et ses limitations si besoin).

2 Chaines de caractères avec Awk

- ▶ Examinez la commande `stat` du *bash* sur un fichier ou répertoire.
- ▶ Réalisez **une ligne de commande** qui pour un fichier/répertoire donné, affichera le nom du fichier/répertoire et sa date d’accès et de modification écrite de cette manière : **Fichier** accédé le JJ/MM/AA à nnH nnMin et nnS. Vous utiliserez *awk* pour récupérer l’information dans la sortie de `stat` ainsi que pour segmenter l’horaire (voir les fonctions de traitement de chaines de caractères, notamment la fonction *split* de *awk*). L’enchaînement des commandes se fera par |.

Aide :

- 🔗 Il y a plusieurs lignes pour l’accès. Il faut prendre celle qui contient Access mais pas Uid

- ▶ Réalisez un script *bash* qui affichera **Fichier** accédé le JJ/MM/AA à nnH nnMin et nnS et modifié le ... pour tous les fichiers/répertoires contenus dans le répertoire courant.