TP1 : Gestion de la mémoire

### Exercice 1

#### Question 1

Quelles sont les différentes options possibles de la commande uname ? Quelles sont les

informations fournies ? Que fournit la commande arch ?

uname :

-a, --all

print all information, in the following order, except omit -p

and -i if unknown:

-s, --kernel-name

print the kernel name

-n, --nodename

print the network node hostname

-r, --kernel-release

print the kernel release

-v, --kernel-version

print the kernel version

-m, --machine

print the machine hardware name

-p, --processor

print the processor type (non-portable)

-i, --hardware-platform

print the hardware platform (non-portable)

-o, --operating-system

print the operating system

--help display this help and exit

--version

output version information and exit

La commande “arch” fournit l’architecture processeur de l’ordinateur (ex x86\_64)

#### Question 2

La commande “uptime” affiche l’heure, le temps de fonctionnement de l’ordinateur, le nombre d’utilisateur connectés et la charge moyenne du processeur.

#### Question 3

cat /proc/sys/kernel/hostname

Retourne le hostname du PC.

grep cpu /proc/stat

Cette commande affiche les différents coeurs du processeur et chaque colonne correspond à ces informations :

* user: normal processes executing in user mode
* nice: niced processes executing in user mode
* system: processes executing in kernel mode
* idle: twiddling thumbs
* iowait: waiting for I/O to complete
* irq: servicing interrupts
* softirq: servicing softirqs

more /proc/loadavg

Cette commande affiche l’utilisation moyenne du processeur à 1, 5 et 15 minutes.

Le 4e champ indique : The first of these is the number of currently executing kernel scheduling entities (processes, threads); this will be less than or equal to the number of CPUs. The value after the slash is the number of kernel scheduling entities that currently exist on the system.

Le 5e champ : PID du processus le plus récent.

cat /proc/uptime

The first value represents the total number of seconds the system has been up. The second value is the sum of how much time each core has spent idle, in seconds.

cat /proc/version

Specifies the version of the Linux kernel, the version of gcc used to compile the kernel, and the time of kernel compilation. It also contains the kernel compiler's user name.

cat /proc/sys/fs/file-max

maximum number of file-handles that the Linux kernel

#### Question 4

Le nombre de coeurs :

cat /proc/cpuinfo | grep "cpu cores" | uniq | awk '{print $NF}'

Le nombre de thread :

cat /proc/cpuinfo | grep processor | c -lw

Si les processeurs sont dual ou multi cores :

Flemme

Si les processeurs sont 64 bit :

uname -i | grep 64 | wc -l

#### Question 5

Il faut être root pour éditer ce fichier.

### Exercice 2

#### Question 1

ps -e -o cmd -o vsize -o size -o pmem

(-e pour afficher un peu plus de processus)

#### Question 2

La commande free affiche l’utilisation de la RAM et de la swap.

watch -n 5 free -m

#### Question 3

cat /proc/meminfo | grep MemFree | awk '{print $2}'

#### Question 4

Ce fichier garde une trace d'une variété de statistiques différentes sur le système depuis son dernier redémarrage.

#### Question 5

procs

Rapports sur les éléments suivants :

* r : La « runqueue » : file de thread prêt à être (ou déjà) en mode « run » - en attente de ressource cpu. Appelée aussi la charge système générale (load average)
* b : La « blocked queue » (ou wait queue) : nombre de thread en attente de ressource autre que la cpu. Ex : disque, mémoire, réseau, etc.

memory

Rapports sur l'utilisation de la mémoire :

* swapd : Espace de swap utilisé
* free : Mémoire disponible
* buff : Mémoire utilisée en buffer
* cache : Mémoire utilisée en cache

swap

Rapports sur l'utilisation de la swap :

* si : « Swap In » : Mémoire déchargée du SWAP vers la RAM par seconde
* so : « Swap Out » : Mémoire déchargée de la RAM vers la SWAP par seconde

io

Rapports sur l'utilisation du disque :

* bi : « Block In » : Lecture sur les périphériques blocs (disques durs) en blocs/s
* bo : « Block Out » : Écriture sur les périphériques blocs (disques durs) en blocs/s

cpu

Rapports sur l'utilisation du processeur :

* us : « Temps Utilisateur » : Opérations lancées en mode utilisateur : les programmes applicatifs, calcul, etc.
* sy : « Temps Système » : Opérations lancées en mode kernel : création de socket, ouverture de fichier, etc.
* id : « Idle » : Temps où le processeur ne fait rien
* wa : « Wait» : En attente de ressource (disque, réseau...)

### Exercice 3 (les réponses sont vagues il faut améliorer)

#### Question 1

Dans “align.c” les variables sont créé dans l’ordre des tailles, pas dans “nonalign.c”

#### Question 2

Vu que c’est pas dans l’ordre bah ça prend plus de place car c’est pas rangé dans la mémoire.

Si on suppose que chaque case de la mémoire est suffisante pour remplir un integer/2 short, si on met d’abord un short, un int puis un short, on remplit 3 cases car une case n’a pas assez de place pour stocker à la fois un short et un int en entier. Alors que dans le cas de figure où on stocke d’abord un int puis deux short à la suite, on remplit une case mémoire pour le int et une case pour les deux shorts.

Il y a donc plus de fragmentations internes dans le premier cas de figure que la deuxième.

C’est l'ordre de déclaration qui change.

#### Question 3

Il faudra plus de plages mémoire dans “nonalign”

#### Question 4

J’ai pas de remarque.

#### Question 5

### Exercice 4