

TP n°1 : Gestion de la mémoire

Exercice 1 :

1. **cat /proc/meminfo**

memtotal : quantité total de Mémoire centrale (MC)
memfree : qte (quantité) de MC disponible
memavailable : une estimation de la quantité de MC qui peut être utilisée par de nouveau processus (mémoire disponible (memfree) + la qte de MC qui peut être récupérée ou libérés)
cached : qte de MC utilisée par linux pour mettre en cache des données fréquemment utilisées
buffers : qte de MC utilisée par linux, les données stockées dans cette mémoire sont destinées à être rapidement accessibles (principe du buffer revoir le cours)
swpcached : qte de données déplacés de la MC à swap mais reste toujours maintenu prêtes à être réutilisés
swaptotal : taille total de la mémoire swap
swapfree : qte de mémoire swap non utilisée
active : qte de MC actuellement utilisées par des processus actif (mémoire récemment utilisée)
inactif : qte de MC qui n'est pas utilisées par des processus actif (mémoire pas récemment utilisée)
kernelstack : qte de MC utilisée pour les piles par les threads niveau kernel
pagetables : qte de MC utilisée par les tables de pages
mapped : qte de MC utilisée pour mappé des fichiers ouvert par des processus

2. Exécutez la commande : **free**

MC : total, used(utilisée), free(disponible), shared : qte de mémoire qui peut être partagées entre plusieurs processus (accessible par plusieurs processus)
Buff/cached, available (memavailable)
Swap : total, used (utilisée), free(disponible)

3. pour obtenir un nouvel affichage toutes les 5 secondes : **watch -n 5 free**

watch : permet d'exécuter une commande et d'actualiser ces résultats chaque période de temps
-n 5 spécifie l'intervalle de temps

4. Pour afficher sur une seule ligne la quantité de mémoire disponible sur le système : **grep MemAvailable /proc/meminfo | awk '{print \$2}'**

grep MemAvailable recherche la ligne qui contient le mot MemAvailable dans le fichier meminfo, une fois la ligne retrouvée, awk recherche la deuxième colonne de cette ligne et affiche son contenu
Awk (utilisé pour traiter et manipuler des fichiers textes)

5. Exécutez la commande : **vmstat**

Process : r (nombre de processus en cours d'utilisation), b (nombre de processus bloqué (en attente))
Memory : swpd : qte de mémoire swap utilisée, free : qte de MC libre, buff : qte de MC utilisée pour le buffer, cache : qte de MC utilisée pour le cache
Swap : si,so : qte de mémoire lus et écrites dans le swap (swap in, swap out)
Io : bi,bo : informations sur les entrées/sorties (input/output)
System : in : nombre d'interruption par seconde, cs : nombre de commutations de contextes par seconde
CPU : us,sy,id,wa,st : informations sur le CPU

6. Exécutez la commande : **cat /proc/<pid>/status**, que permet-elle d'obtenir ? (Utilisez la commande top pour choisir un processus)

Cette commande fournit des informations sur les processus (l'état actuel du processus)
Nom processus
State processus : exemple running (en cours d'exécution)
Umask : les permissions de lectures, écritures (les droits d'accès)
Tgid : thread group ID, l'identifiant attribué à un groupe de threads associé au processus
Groups : les id des groupes auxquelles appartient le processus
FDsize : taille de la table des descripteurs de fichiers pour un processus (nombre de fichiers ouverts dans la table des descripteurs de fichiers)

Vmsize : taille de la mémoire virtuelle (swap) utilisée par le processus y compris la qte se trouvant dans la mémoire centrale et celle dans le swap

Vmswap : qte de mémoire virtuelle déplacée vers l'espace swap

Sigpnd : signaux pas encor traité (pending, pendant)

Sigblk : signaux bloqués

Sigign : signaux ignorés

7. Exécutez **pmap <PID>**

Cette commande affiche la carte mémoire pour le processus spécifié avec son PID

Adresse : adresse de début de la région mémoire au processus

Taille : taille de la region

Permission d'accès (droits d'accès)

Path : nom fichier ou chemin fichier associé a cette region (cela veut dire que le contenu de cette region est chargé a partir du fichier spécifié)

Remarque : anon représente des allocations dynamique de mémoire effectués par le processus

8. ensuite **cat /proc/<PID>/maps**

Cette commande affiche aussi la carte mémoire pour le processus spécifié avec son PID

Plage d'adresse pour cette region de mémoire dans le processus spécifié

Droits d'accès

Décalage (offset) par rapport au début du fichier si la region a été mappé a partir d'un fichier

Dispositif (infos sur le périphérique) a partir du quel la region mémoire a été mappé

Inode (numéro de l'inode associé au fichier) si la région a été mappé a partir d'un fichier

Nom du fichier/dispositif associé a la région mémoire (chemin du fichier ou dispositif)

Les commande **pmap** et **/proc/<PID>/maps** permettent d'obtenir des informations sur la carte mémoire du processus spécifié par PID qui montrent les différentes régions du processus