

TP N°2 : Control groups « cgroups » – 3exercices – Durée approx. ~ 1h30min.

A l'aide du support de cours et des éventuels mémentos réalisez les exercices suivants :

(Tout code ou implémentation compilant ou non sera étudié)

Le présent sujet de TP comporte 3 pages

I – Limiter la vitesse d'accès disque

Le but de cet exercice est de mettre en œuvre un cgroup limitant la vitesse d'I/O disque plafonnant la vitesse de lecture d'un processus spécifique.

La démarche se déroule en trois temps à savoir :

1. *La réalisation d'un benchmark disque.*
 2. *La mise en place du cgroup sur le sous-système adéquat.*
 3. *Le constat de l'exécution de la contrainte mise en place.*
- Pour réaliser le benchmark disque il est nécessaire de passer par la commande **dd**.
 - Réalisez deux tests consécutifs générant deux fichiers de 2Go chacun et notez le taux de transfert.
 - Pour les paramètres de la commande aidez-vous du manuel (piste: `if=/dev/urandom` ou `/dev/random`).
 - Mettez en place un cgroup nommé **tp1_blkio** portant sur le sous-système **blkio** à l'aide des commandes de base d'un terminal (**cgcreate** interdit). Ce dernier doit limiter la vitesse de lecture à 5mo/s
 - Référez-vous à la section **blkio.throttle.xxx**.
 - (Concernant les numéros majeurs/mineurs documentez-vous sur la commande **ls -l /dev/sdX**).
 - Pour réaliser un nouveau benchmark et vérifier la prise en compte utilisez les commandes **iostat** et **dd** au sein deux terminaux distincts.
 - Lancez **iostat** au sein d'un premier terminal.
 - Lancez la commande **dd** au sein d'un second terminal en vous appuyant sur la commande **cgexec**.
 - Pour les paramètres de la commande aidez-vous du manuel (piste : `-g blkio:tp1_blkio`).
- ❖ *Que constatez-vous ?*
- ❖ *Avec deux terminaux exécutant la commande **dd** ?*

II – Partager la mémoire

Le but de cet exercice est de mettre en œuvre un cgroup limitant la quantité de mémoire disponible pour un spécifique.

La démarche se déroule en trois temps à savoir :

1. *La réalisation d'un programme allouant un certain montant de mémoire.*
 2. *La mise en place du cgroup sur le sous-système adéquat.*
 3. *Le constat de l'exécution de la contrainte mise en place.*
- Pour réaliser la consommation de mémoire, réalisez un utilitaire écrit en C réalisant des appels successifs à la fonction **malloc** prenant en paramètre le nombre de méga-octets à allouer (ie : `./mem_conso 50`) et affichant l'état courant sur la sortie standard.
 - Mettez en place un cgroup nommé **tp1_mem** portant sur le sous-système **memory** à l'aide des commandes de base d'un terminal (**cgcreate** interdit). Ce dernier doit limiter l'usage mémoire à 50Mo.
 - Référez-vous à la section **memory.xxx**.

- Pour réaliser la vérification de la prise en compte de la contrainte utilisez votre utilitaire **mem_conso** et **htop** au sein de deux terminaux distincts.
 - Lancez **htop** au sein d'un premier terminal.
 - Lancez votre utilitaire d'un second terminal après avoir injecté son **pid** au sein du fichier **cgroup/memory/tp1_mem/tasks**.
 - Pour les paramètres de la commande aidez-vous du manuel (piste : **-g blkio:tp1_blkio**).
- Que constatez-vous ?

III – “Clean-up that mess !”

L'objectif de cet exercice est de réaliser le nettoyage des cgroups créés précédemment. Néanmoins vous allez sauvegarder la configuration de ces derniers afin de pouvoir la réinjecter au prochain démarrage.

- A l'aide de la commande **cgsnapshot** sauvegardez la configuration des groupes.
- A l'aide des commandes **cgdelete** et **lscgroup** supprimez et vérifiez que les groupes sont effectivement détruits.
- Faites en sorte de rendre la configuration permanente pour l'utilisateur **cnam** à l'aide des fichiers **/etc/cgconfig.conf** et **/etc/cgrules.conf** vérifiez la prise en compte à l'aide du mode opératoire de l'exercice N°1 ou N°2 ainsi que la commande **lscgroup**.

/!\ Pensez à vider le cache mémoire avant chaque **dd** : `sync ; echo 3 > /proc/sys/vm/drop_caches`

IV – Rappels

Approche incrémentale du développement

Pour obtenir les résultats attendus aux différents exercices, veuillez toujours appliquer une approche incrémentale en termes d'ajout de code/fonctionnalité.

Par exemple: une approche incrémentale pour ce type d'exercice "Is-like" serait :

1. La récupération des paramètres.
2. Tester fichier/répertoire.
3. Parcourir les éléments du répertoire en affichant leur nom.
4. Alimenter chaque fichier avec une information supplémentaire : permission / taille / propriétaire...

Documentation

Pour obtenir des informations ou de la documentation ayez le réflexe d'utiliser les pages du manuel.

Par exemple:

➤ man 3 stat / man 2 open / man 2 readdir / man errno

Gestion des erreurs

Afin d'avoir une gestion des erreurs la plus précise possible ayez le réflexe d'utiliser les codes retours **ERRNO** spécifiés dans les pages de manuel

Par exemple:

- | | |
|------------------|---|
| ➤ EEXIST | File exists (POSIX.1) |
| ➤ EFAULT | Bad address (POSIX.1) |
| ➤ EISDIR | Is a directory (POSIX.1) |
| ➤ ENOTDIR | Not a directory (POSIX.1) |
| ➤ ELOOP | Too many levels of symbolic links (POSIX.1) |