

TP AISE — Conteneurs / VMs

Exercice 1 : I/Os Structurées

Le but de cet exercice est de faire des entrées sorties binaires à partir d'une struct en C. Vous devrez stocker un tableau de struct particle de taille NUM_PART (par exemple 64) et le relire.

```
struct particle
{
    int x, y, z;
};
```

Les particules sont initialisées comme suit:

```
for( i = 0 ; i < NUM_PART; i++)
{
    parts[i].x = i;
    parts[i].y = i + 1;
}
```

Les particules sont vérifiées comme suit:

```
for( i = 0 ; i < NUM_PART; i++)
{
    if( (parts[i].x != i) || (parts[i].y != (i + 1)) )
    {
        fprintf(stderr, "Erreur dans le buffer %d\n", i);
        fprintf(stderr, "x %d vs %d\n", parts[i].x, i);
        fprintf(stderr, "y %d vs %d\n", parts[i].y, i);
        return 1;
    }
}
```

Exercice 2 : Redirection de commande

Faite un programme qui en reposant sur un pipe lance l'équivalent des commandes suivantes:

echo "8+4" | bc

Si bc n'est pas présent ou que vous ne parvenez pas à installer « bc » remplacez cette commande par « cat »

Exercice 3 : Lancement de VM qemu

Récupérez la VM (soit via clef USB ou bien via france.paratools.com/chps.qcow2) et lancez la avec le support réseau.

Réalisez ensuite un lancement en -nographic en passant les options compatibles avec le CPU hotplug et le ballooning. Testez ensuite ces configurations dans votre VM avec la console (CTRL+a puis c). Pouvez-vous arrêter la VM depuis la console ?

Notez que cette VM contient Docker pour ceux qui n'arrivent pas à l'installer sur leur machine. Pour avoir docker-compose faire *apt-get install docker-compose*.

Exercice 4 : Injection de répertoire

Dans le conteneur nginx, le serveur web sert une page par défaut. Le but de cet exercice est de changer cette page afin de servir votre propre site web. Vous devrez alors injecter le site fourni en exemple ou bien une page de votre construction (appelée index.html) et la servir via le conteneur.

Pour ce faire, vous devrez:

- Localiser le répertoire racine de nginx (voir fichier de configuration dans /etc/...)
- Monter votre propre dossier en lieu et place de ce répertoire

Exercice 5: Création de Conteneur minimal

Un conteneur contient généralement un certain nombre de bibliothèques indispensables. Cependant, il est possible de créer des conteneurs minimaux avec l'image de base « scratch » qui fournit le minimum vital pour des binaires statiques.

Créez un binaire statique (-static à GCC) affichant un message et ensuite construisez une Dockerfile héritant de « scratch » et lançant ce binaire. Enfin vérifiez qu'il fonctionne.

Exercice 6 : Création d'un conteneur Apache

En vous inspirant de ce qui a été fait pour nginx dans le cours créer un conteneur lançant apache. Il vous faudra trouver l'option qui permet de lancer apache en mode « interactif ». Note vous pouvez tenter de regarder dans l'image apache officielle.

Exercice 7 : Redis

Lancez le conteneur officiel REDIS et connectez-vous au serveur avec netcat. Testez les commandes de base de redis (voir ici <https://redis.io/commands>). Par exemple:

```
SET a 99
GET a
UNLINK a
```

Ensuite implémentez une page web contenant un formulaire permettant d'ajouter un utilisateur à la base de donnée avec un mot de passe qui sera stocké en hash MD5. Dans le même temps implémentez un script python qui consomme régulièrement les utilisateurs de la base de donnée et vérifie que le mot de passe ne peut être cassé de manière triviale (4-5 lettre maximum) ce script python ajoutera un drapeau dans la base de donnée si le mot de passe est cassé. Vous réaliserez enfin une page PHP listant les mots de passes cassés.