Sistemas Operativos
Práctica 4 (I) - cgroups & namespaces

Parte 1: Conceptos teóricos

1. Defina virtualización. Investigue cuál fue la primera implementación que se realizó.
2. ¿Qué diferencia existe entre virtualización y emulación?
3. Investigue el concepto de hypervisor y responda:
    (a) ¿Qué es un hypervisor?
    (b) ¿Qué beneficios traen los hypervisors? ¿Cómo se clasifican?
4. ¿Qué es la full virtualization? ¿Y la virtualización asistida por hardware?
5. ¿Qué implica la técnica binary translation? ¿Y trap-and-emulate?
6. Investigue el concepto de paravirtualización y responda:
    (a) ¿Qué es la paravirtualización?
    (b) Mencione algún sistema que implemente paravirtualización.
    (c) ¿Qué beneficios trae con respecto al resto de los modos de virtualización?
7. Investigue sobre containers y responda:
    (a) ¿Qué son?
    (b) ¿Dependen del hardware subyacente?
    (c) ¿Qué lo diferencia por sobre el resto de las tecnologías estudiadas?
    (d) Investigue qué funcionalidades son necesarias para poder implementar containers.

Parte 2: chroot, Control Groups y Namespaces

Debido a que para la realización de la práctica es necesario tener más de una terminal abierta simultáneamente tenga en cuenta la posibilidad de lograr esto mediante alguna alternativa (ssh, terminales gráficas, etc.)

Chroot

En algunos casos suele ser conveniente restringir la cantidad de información a la que un proceso puede acceder. Uno de los métodos más simples para aislar servicios es `chroot`, que consiste simplemente en cambiar lo que un proceso, junto con sus hijos, consideran que es el directorio raíz, limitando de esta forma lo que pueden ver en el sistema de archivos. En esta sección de la práctica se preparará un árbol de directorios que sirva como directorio raíz para la ejecución de una shell.

1. ¿Qué es el comando chroot? ¿Cuál es su finalidad?
2. Crear un subdirectorio llamado sobash dentro del directorio root. Intente ejecutar el comando chroot /root/sobash. ¿Cuál es el resultado? ¿Por qué se obtiene ese resultado?
3. Copiar en el directorio anterior todas las librerías que necesita el comando bash. Para obtener esta información ejecutar el comando **ldd /bin/bash**. ¿Es necesario copiar la librería linux-vdso.so.1? ¿Por qué? Dentro del directorio anterior crear las carpetas donde va el comando bash y las librerías necesarias. Probar nuevamente. ¿Qué sucede ahora?
4. ¿Puede ejecutar los comandos cd "directorio" o echo? ¿Y el comando ls? ¿A qué se debe esto?
5. ¿Qué muestra el comando pwd? ¿A qué se debe esto?
6. Salir del entorno chroot usando exit

Control Groups

A continuación se probará el uso de cgroups. Para eso se crearán dos procesos que compartirán una misma CPU y cada uno la tendrá asignada un tiempo determinado.
Nota: es posible que para ejecutar xterm tenga que instalar un gestor de ventanas. Esto puede hacer con apt-get install xterm.

1. ¿Dónde se encuentran montados los cgroups? ¿Qué versiones están disponibles?
2. ¿Existe algún controlador disponible en cgroups v2? ¿Cómo puede determinarlo?
3. Analice qué sucede si se remueve un controlador de cgroups v1 (por ej. Umount /sys/fs/cgroup/rdma).
4. Crear dos cgroups dentro del subsistema cpu llamados cpualta y cpubaja. Controlar que se hayan creado tales directorios y ver si tienen algún contenido
   ```
   # mkdir /sys/fs/cgroup/cpu/"nombre_cgroup"
   ```
5. En base a lo realizado, ¿qué versión de cgroup se está utilizando?
6. Indicar a cada uno de los cgroups creados en el paso anterior el porcentaje máximo de CPU que cada uno puede utilizar. El valor de cpu.shares en cada cgroup es 1024. El cgroup cpualta recibirá el 70 % de CPU y cpubaja el 30 %.
   ```
   # echo 717 > /sys/fs/cgroup/cpu/cpualta/cpu.shares
   # echo 307 > /sys/fs/cgroup/cpu/cpubaja/cpu.shares
   ```
7. Iniciar dos sesiones por ssh a la VM.(Se necesitan dos terminales, por lo cual, también podría ser realizado con dos terminales en un entorno gráfico). Referenciaremos a una terminal como termalta y a la otra, termbaja.
8. Usando el comando taskset, que permite ligar un proceso a un core en particular, se iniciará el siguiente proceso en background. Uno en cada terminal. Observar el PID asignado al proceso que es el valor de la columna 2 de la salida del comando.
   ```
   # taskset -c 0 md5sum /dev/urandom &
   ```
9. Observar el uso de la CPU por cada uno de los procesos generados (con el comando top en otra terminal). ¿Qué porcentaje de CPU obtiene cada uno aproximadamente?
10. En cada una de las terminales agregar el proceso generado en el paso anterior a uno de los cgroup (termalta agregarla en el cgroup cpualta, termbaja en cpubaja. El process_pid es el que obtuvieron después de ejecutar el comando taskset)
    ```
    # echo "process_pid" > /sys/fs/cgroup/cpu/cpualta/cgroup.procs
    ```

11. Desde otra terminal observar cómo se comporta el uso de la CPU. ¿Qué porcentaje de CPU recibe cada uno de los procesos?
12. En termalta, eliminar el job creado (con el comando jobs ven los trabajos, con kill %1 lo eliminan. No se olviden del %.). ¿Qué sucede con el uso de la CPU?
13. Finalizar el otro proceso md5sum.
14. En este paso se agregarán a los cgroups creados los PIDs de las terminales (Importante: si se tienen que agregar los PID desde afuera de la terminal ejecute el comando echo $$ dentro de la terminal para conocer el PID a agregar. Se debe agregar el PID del shell ejecutando en la terminal).
    ```
    # echo $$ > /sys/fs/cgroup/cpu/cpualta/cgroup.procs (termalta)
    # echo $$ > /sys/fs/cgroup/cpu/cpubaja/cgroup.procs (termbaja)
    ```
15. Ejecutar nuevamente el comando taskset -c 0 md5sum /dev/urandom & en cada una de las terminales. ¿Qué sucede con el uso de la CPU? ¿Por qué?
16. Si en termbaja ejecuta el comando: taskset -c 0md5sum /dev/urandom & (deben quedar 3 comandos md5 ejecutando a la vez, 2 en el termbaja). ¿Qué sucede con el uso de la CPU? ¿Por qué?

Namespaces

1. Explique el concepto de namespaces.
2. ¿Cuáles son los posibles namespaces disponibles?
3. ¿Cuáles son los namespaces de tipo Net, IPC y UTS una vez que inicie el sistema (los que se iniciaron la ejecutar la VM de la cátedra)?
4. ¿Cuáles son los namespaces del proceso cron? Compare los namespaces net, ipc y uts con los del punto anterior, ¿son iguales o diferentes?
5. Usando el comando unshare crear un nuevo namespace de tipo UTS.
    a. unshare –uts sh (son dos (- -) guiones juntos antes de uts)
    b. ¿Cuál es el nombre del host en el nuevo namespace? (comando hostname)
    c. Ejecutar el comando lsns. ¿Qué puede ver con respecto a los namespace?.
    d. Modificar el nombre del host en el nuevo hostname.
    e. Abrir otra sesión, ¿cuál es el nombre del host anfitrión?
    f. Salir del namespace (exit). ¿Qué sucedió con el nombre del host anfitrión?
6. Usando el comando unshare crear un nuevo namespace de tipo Net.
    a. unshare –pid sh
    b. ¿Cuál es el PID del proceso sh en el namespace? ¿Y en el host anfitrión?
    c. Ayuda: los PIDs son iguales. Esto se debe a que en el nuevo namespace se sigue viendo el comando ps sigue viendo el /proc del host anfitrión. Para evitar esto (y lograr un comportamiento como los contenedores), ejecutar:
        unshare --pid --fork --mount-proc
    d. En el nuevo namespace ejecutar ps -ef. ¿Qué sucede ahora?
    e. Salir del namespace