



Lab 11.1: El sistema de archivos especial tmpfs

tmpfs es uno de muchos sistemas de archivos especiales usados en **Linux**. Algunos de estos no son usados realmente como sistemas de archivos, pero toman ventaja de la capa de abstracción que poseen. Sin embargo, **tmpfs** es un sistema de archivos real sobre el cual las aplicaciones pueden realizar operaciones de E/S.

Esencialmente, **tmpfs** opera como **ramdisk**; reside completamente en memoria. Pero tiene algunas características interesantes que las implementaciones convencionales y antiguas de ramdisk no tenían:

1. El sistema de archivos ajusta su tamaño (The filesystem adjusts its size (y por lo tanto la memoria que se usa) dinámicamente; parte en cero y se expande tanto como sea necesario hasta el tamaño máximo de la partición en la que está montada.
2. Si la RAM se agota, **tmpfs** puede utilizar espacio del área de intercambio. De todas formas no es posible poner más en el sistema de archivos de la capacidad máxima que soporta.
3. **tmpfs** no requiere tener un sistema de archivos normal, tales como **ext3** o **vfat**; posee sus métodos propios para lidiar con archivos y operaciones de E/S, los cuales están conscientes de que es sólo espacio en memoria (y que en realidad no es un dispositivo de bloque), y como tal está optimizado para velocidad.

Por lo tanto no hay necesidad de preformatear el sistema de archivos con el comando **mkfs** ; simplemente hay que montarlo y usarlo.

Monte una instancia nueva de **tmpfs** en cualquier lugar en su estructura de directorios, con un comando como el siguiente:

```
$ sudo mkdir /mnt/tmpfs
$ sudo mount -t tmpfs none /mnt/tmpfs
```

Vea cuánto espacio se le ha asignado al sistema de archivos y cuánto está usando:

```
$ df -h /mnt/tmpfs
```

Debería notar que se le ha asignado un valor por defecto de la mitad de la RAM del sistema; sin embargo, el uso es cero, y sólo comenzará a crecer en espacio utilizado en la medida de que se ponen archivos en `/mnt/tmpfs`.

Es posible cambiar el tamaño asignado como una opción a la hora de montarlo:

```
$ sudo mount -t tmpfs -o size=1G none /mnt/tmpfs
```

Podría intentar llenarlo hasta que se alcance la capacidad máxima y luego vea qué sucede. No olvide desmontarlo una vez que haya terminado las pruebas, con el comando:

```
$ sudo umount /mnt/tmpfs
```

Prácticamente todas las distribuciones modernas de **Linux** montan una instancia de **tmpfs** en `/dev/shm`:

```
$ df -h /dev/shm
```

```
Filesystem      Type      Size  Used Avail Use% Mounted on
tmpfs           tmpfs     3.9G   24M  3.9G   1% /dev/shm
```

Muchas aplicaciones hacen esto en casos como cuando se usa memoria compartida **POSIX** como un mecanismo de comunicación interprocesos. Cualquier usuario puede crear, leer y escribir archivos en `/dev/shm`, por lo que es un buen lugar para crear archivos temporales en memoria.

Cree algunos archivos en `/dev/shm` y observe con **df** cómo el sistema de archivos se va llenando.

Adicionalmente, muchas distribuciones montan instancias múltiples de **tmpfs**; por ejemplo, se observa lo siguiente en un sistema **RHEL 7**:

```
$ df -h | grep tmpfs
```

```
devtmpfs                devtmpfs  3.9G      0  3.9G    0% /dev
tmpfs                   tmpfs     3.9G    24M  3.9G    1% /dev/shm
tmpfs                   tmpfs     3.9G    9.2M  3.9G    1% /run
tmpfs                   tmpfs     3.9G      0  3.9G    0% /sys/fs/cgroup
/tmp/vmware-coop/564d9ea7-8e8e-29c0-2682-e5d3de3a51d8 tmpfs     3.3G      0  3.3G    0% /tmp/vmware-coop/
564d9ea7-8e8e-29c0-2682-e5d3de3a51d8
/tmp/vmware-coop/564d7668-ec55-ee45-f33e-c8e97e956190 tmpfs     2.3G    2.0G  256M   89% /tmp/vmware-coop/
564d7668-ec55-ee45-f33e-c8e97e956190
none                    tmpfs     1.0G    1.0G      0 100% /tmp/ohno
```

Note que el comando anterior fue ejecutado en un sistema con 8GB de RAM, por lo cual usted probablemente no tendrá todos esos sistemas de archivos **tmpfs** usando los 4 GB con los cuales han sido asignados.

Algunas distribuciones como **Fedora** podrían montar por defecto `/tmp` como un sistema **tmpfs**; en estos casos uno podría evitar poner archivos grandes en `/tmp` para que el sistema no se quede sin memoria. Otra posibilidad es deshabilitar ese comportamiento, como se mencionó anteriormente al describir `/tmp`.