Departamento de Cs. e Ingeniería de la Computación
Universidad Nacional del Sur

Sistemas Operativos y Distribuidos

Trabajo Práctico N° 8
Sistemas de Archivos

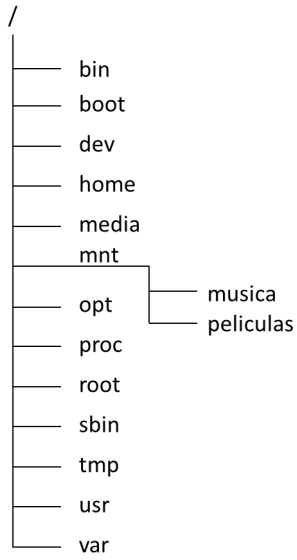
Segundo Cuatrimestre de 2015

Ejercicios

1. Hemos visto que para el mismo *kernel* de Linux pueden existir múltiples distribuciones, cada una equipada con su propio software y filosofía. Más aún, para cada distribución pueden existir múltiples *file systems* que pueden ser escogidos y utilizados por el usuario, incluso de forma simultánea en diversas particiones. ¿Bajo qué concepto la estructura de la jerarquía de directorios se mantiene invariable independientemente de la distribución o el sistema de archivos utilizado?

2. ¿Qué entiende por *punto de montaje* o *mount point*? ¿Cuál es el propósito de los archivos */etc/fstab* y */etc/mtab*?

3. Dada la siguiente jerarquía en el contexto de los sistemas operativos Unix, y en particular GNU/Linux,



/etc/fstab

```
/dev/sda3  /          ext4     defaults     1 1
/dev/sda4  /home      ext3     defaults     1 2
/dev/sda6   swap      swap     defaults     0 0
 tmpfs     /tmp       tmpfs    defaults     0 0
/dev/sda8  /media/W   ntfs-3g  noauto,user  0 0
```

   a) ¿Qué particiones y sistemas de archivos existen en el sistema?

   b) ¿Qué particiones y sistemas de archivos son montados cuando inicia el sistema?

   c) ¿Cuál es el objetivo de los directorios *musica* y *peliculas*?

d) ¿Cuál es el efecto de la ejecución de los siguientes comandos?

```
$ mount -t ntfs -o ro /dev/sda9 /mnt/musica/
$ mount /media/W
```

¿Por qué en el segundo caso sólo se especifica el destino del punto de montaje?

e) Suponga que un usuario desea copiar un archivo desde el directorio *musica* a su directorio *home*. Para esto, ejecuta el comando,

```
$ cp /mnt/musica/triunfo.mp3  /mi_musica/
```

¿La ejecución del comando tiene éxito? Tenga en cuenta que en el proceso existen ***dos sistemas de archivos distintos*** involucrados. Asuma que el usuario posee los permisos suficientes para realizar la operación.

4. Explique cómo administran los sistemas Windows el *espacio de nombres* que modela la información de los diversos sistemas de archivos utilizados. ¿Cuál es la diferencia con los sistemas Unix?

5. ¿Cuál es el objetivo del sistema de archivos */proc*? Explique el resultado de la ejecución del siguiente comando:

```
# echo "1" >/proc/sys/net/ipv4/ip_forward
```

6. Implemente un programa denominado *fileinfo* el cual permita obtener la información referente al archivo cuyo nombre se pasa como parámetro. A continuación se muestran dos ejemplos del funcionamiento del programa:

```
$ fileinfo fileinfo.c
 Archivo: fileinfo
 Número de i-nodo: 1049918
 Tipo: Archivo convencional                    Tamaño de archivo: 12393 bytes
 Tamaño de bloque: 4096 bytes                  Bloques asignados: 32
 UID: 1000 (sosd)                              GID: 1001 (sosd)
 Permisos: rwxr-xr-x
 Último acceso: Thu 20 20:10:31 2011
 Última modificación: Thu Oct 20 20:10:30 2011

$ fileinfo /dev/sda

  Archivo: /dev/sda
  Número de i-nodo: 3063
  Tipo: Block device                           Tamaño de archivo: 0 bytes
  Tamaño de bloque: 4096 bytes                 Bloques asignados: 0
  UID: 0 (root)                                GID: 6 (disk)
  Permisos: rw-rw-
  Último acceso: Thu 20 20:06:58 2011
  Última modificación: Thu Oct 20 18:04:09 2011
```

7. Considere un *file system* en el cual los archivos son almacenados en bloques de 16 KB. Ignorando la sobrecarga debida a la administración de directorios y descriptores, calcule el nivel de fragmentación interna para cada uno de los siguientes tamaños de archivo:

   a) 41600 bytes

   b) 640000 bytes

   c) 4064000 bytes

8. En un determinado sistema del tipo Unix, la longitud de bloque es de 1 KB y cada bloque puede almacenar hasta un total de 256 direcciones. Calcule el tamaño máximo de un archivo asumiendo el esquema tradicional de *i-nodos* (indirección simple, doble y triple).

9. Un sistema de archivos utiliza bloques físicos con un tamaño de 256 bytes y un mecanismo de *asignación enlazada*. Cada archivo posee una entrada de directorio la cual mantiene información sobre el nombre del archivo, la ubicación del primer bloque, la longitud y la posición del último bloque. Asumiendo que la entrada de directorio se encuentra en memoria, especifique cuántos bloques físicos deben leerse para acceder a la ubicación del bloque requerido en cada uno de los siguienes casos:

   a) Último bloque leído: 100; bloque a leer: 600

   b) Último bloque leído: 500; bloque a leer: 200

   c) Último bloque leído: 20; bloque a leer: 21

   d) Último bloque leído: 21; bloque a leer: 20

10. Suponga que la estructura utilizada para mantener los bloques libres en disco se pierde completamente debido a una caída inesperada del sistema. ¿Existe alguna forma de recuperar esta información? En caso de que su respuesta sea afirmativa, explique brevemente cómo implementaría dicho mecanismo tanto en un sistema del tipo Unix como en un sistema que utilice una tabla FAT.

11. En muchos sistemas Unix se ha sugerido que la primera parte de un archivo sea almacenada en el mismo bloque de disco que su *i-nodo*. ¿Cuál es la razón de este planteo y cuál sería la ventaja de este enfoque?

12. Un sistema posee un disco con 1000 cilindros, numerados desde 0 a 999. En un determinado momento la cola de requermientos posee la siguiente información: 123, 874, 692, 475, 105 y 376. Compute el número de tracks que deben ser recorridos en los siguientes casos, si el útlimo requerimiento atendido fue el 345 y la cabeza del disco se mueve hacia el track 0.

    a) FIFO

    b) SSTF

    c) SCAN

    d) LOOK

    e) C-SCAN

    f) C-LOOK

13. ¿El algoritmo SSTF favorece a ciertos bloques de disco dependiendo de su ubicación?

14. En un determinado momento la cola de requerimientos pendientes es

$$27, 129, 110, 186, 147, 41, 10, 64, 120$$

Suponiendo que el cabezal del disco inicialmente está posicionado sobre el track 100 y se mueve en dirección ascendente con respecto al número de track, determine

a) Cada track accedido

b) Número de tracks recorridos

c) Longitud de búsqueda (seek) promedio

Para los algoritmos de planificación FIFO, SSTF, SCAN y C-SCAN asumiendo que el cabezal del disco se mueve en dirección decreciente con respecto al número de track. Repita el análisis considerando que el cabezal del disco se mueve en dirección contraria.

15. Un disco de 7200 rpm posee 26310 cilindros, 16 cabezales y 63 sectores por *track*. El tiempo de *seek* entre tracks adjactentes es de 1 ms. Detemine el tiempo para leer la totalidad del disco si la cabeza lecto-escritora está posicionada sobre el track 0.

16. Algunos sistemas de archivos como *ext4* o *Reiser4* soportan el concepto de *extents*. Básicamente, un archivo se estructura como una colección de extents, donde cada extent corresponde a un conjunto de bloques contiguos. Un aspecto clave en este tipo de sistemas corresponde a definir correctamente el tamaño de los extents. Indique las ventajas y desventajas de cada uno de los siguientes esquemas:

a) Todos los extents poseen el mismo tamaño predeterminado.

b) Los extents pueden tener distintos tamaños y son alocados dinámicamente.

c) Los extents pueden poseer un conjunto reducido tamaños predeterminados.

17. Suponga que desea comparar un sistema tradicional (sin RAID) con un sistema que soporte RAID.

a) ¿Qué niveles de RAID ofrecerían una mejora de performance ante la mayoría de las cargas típicas?

b) ¿Cambiaría su respuesta en el caso de que se desee priorizar la confiabilidad del sistema?

18. Explique cuáles serían las implicancias de mantener la semántica de consistencia Unix en un sistema de archivos distribuido.

19. ¿Cuáles son los beneficios de un sistema de archivos distribuido con respecto a un sistema de archivos centralizado?

20. Considere un sistema distribuido en el cual el sistema de archivos utiliza un espacio de nombres sustentado de acuerdo al nombre asociado a cada usuario del sistema. En otras palabras, cada usuario posee su propio espacio de nombres privado. ¿Pueden utilizarse los nombres contenidos en dicho sistema para compartir recursos entre usuarios?