

# Sistemas Operativos - laboratorio 2

Profesor: Cristobal Acosta

Ayudantes: Adolfo Diaz & Pablo Gonzalez C.

Segundo semestre 2014

## 1 Objetivo

Aprender a trabajar con procesos en C/C++.

## 2 Objetivos Específicos

- Creación de procesos en Linux.
- Utilización de GetOpt.
- Utilización de pipes para la comunicación entre procesos.

## 3 Enunciado

RAID (*redundant array of independent disks*) es una tecnología de virtualización de almacenamiento de datos en la cual se combinan múltiples unidades de disco en una unidad lógica con la finalidad de obtener redundancia de datos y/o mejora en el rendimiento del sistema.

Existen varios esquemas de configuración de discos al momento de realizar un RAID, para este laboratorio se utilizaran las combinación de dos de ellos.

- RAID 0: Se realizan mediante la distribución de bloques que se guardan en los discos que componen el sistema, este nivel no proporciona redundancia de datos o tolerancia a fallas pero si mejora el rendimiento a través de la lectura y escritura en múltiples unidades.
- RAID 1: Tambien conocido como espejo, los datos se escriben en forma idéntica en una o mas unidades de disco, produciendo un conjunto duplicado de datos. La lectura es atendida por cualquiera de los discos que componen el RAID. Otorga tolerancia a fallas.

### 3.1 Solución a implementar

La solución a que se pide implementar es una simulación de un RAID 0+1 (también llamado RAID 01) que tiene las características de ser utilizado tanto para replicar los datos y poseer tolerancia a fallas como para compartir los datos en distintos discos y obtener mayor rendimiento.

La configuración de los discos y su subdivisión de archivos se puede apreciar en al figura 1, es necesario destacar que la implementación no se debe limitar al número mínimo de discos dando la posibilidad de lograr nuevas configuraciones tal como se puede observar en la Figura 2.

Para realizar la implementación se pide que cada disco se simule a nivel de software a través de un proceso, y la comunicación entre ellos se realice a través de pipes. A la vez se debe crear un proceso adicional que realizara el trabajo de una controladora RAID la cual estará encargada del manejo y recuperación de archivos.

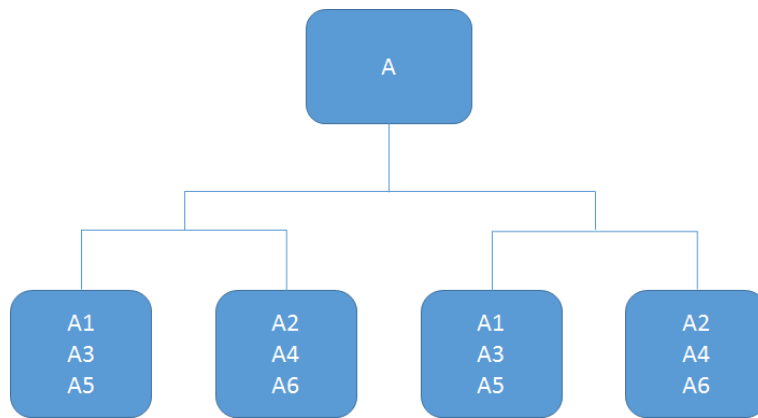


Figura 1: Ejemplo configuracion RAID 0+1.

## 4 Especificaciones

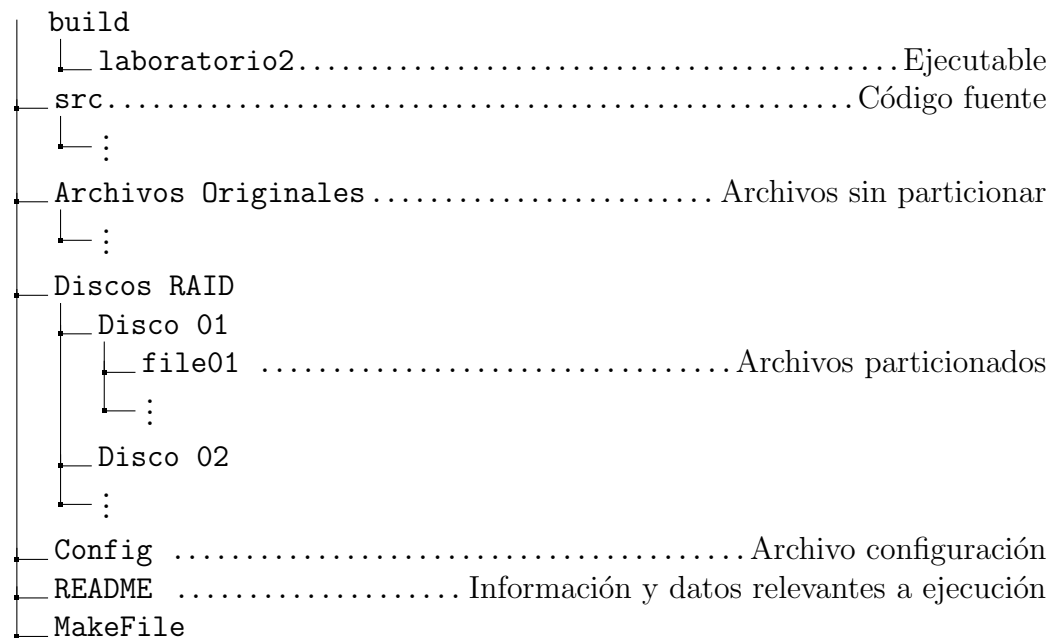
- La implementación se debe realizar a través de procesos.
- La comunicación entre procesos se debe realizar exclusivamente a través de pipes.
- El tamaño del bloque debe ser de 1024 bytes.
- Se debe trabajar con archivos de configuración y almacenamiento para que la solución a implementar sea persistente.
- Se debe entregar el programa con un MakeFile.
- El archivo binario se debe llamar laboratorio2.
- El programa debe ser subido a Usach Virtual en un archivo .tar con el rut del alumno, se aceptan envíos con hasta 24 horas de atraso con nota máxima un 4, cualquier entrega superior a este umbral será clasificado con nota mínima.
- Las pruebas se realizarán en los computadores del laboratorio de informática.

## 4.1 Banderas y funcionamiento

- Crear un sistema de almacenamiento que utilice RAID (-C):
  - Recibe un número que indica la cantidad de discos que ha utilizar en la nueva configuración RAID, en caso que ya exista un sistema de almacenamiento creado, no se debe permitir la creación de otro.
- Eliminar RAID (-E)
  - Se realiza la eliminación de todo el contenido de la carpeta *Discos RAID* y el archivo de configuración.
- Guardar archivo en RAID (-g)
  - La bandera -g recibe el nombre del archivo a guardar en los discos RAID.
- Eliminar bloque aleatorio en RAID (-e)
  - Recibe el nombre de archivo al que se le borrara un bloque aleatorio.
- Restaurar archivo faltante (-r)
  - Recibe el nombre del archivo al cual se le intentara restaurar los archivos faltantes.
- Recuperar archivo de los discos en RAID (-R)
  - Recibe el nombre de archivo a obtener del sistema de almacenamiento y lo guardar en la carpeta *archivos originales*.
- Mostrar archivos (-m)
  - Archivos sin errores
    - \* Archivos que tengan todos los bloques en los discos correspondientes
  - Archivos corruptos
    - \* Archivos que le falten uno o varios bloques, pero deben tener una duplica.

- Archivos con bloques inexistentes en el original y en el respaldo
  - \* Falta el mismo bloque en el respaldo y en el original.

## 4.2 Organización Carpetas



## 4.3 Makefile

Se debe crear un Makefile que contenga dos reglas como mínimo:

- Por defecto: Compila y crea el ejecutable.
- clean: Eliminar el archivo binario creado con anterioridad.

## 5 Validación

Las banderas deben tener las siguientes validaciones y avisos por pantalla:

- -C:
  - Si ya existe un sistema de almacenamiento se debe dar aviso y no realizar nada.
  - En caso de que la operación sea satisfactoria se debe realizar y avisar por pantalla.
- -E:
  - Si no existe un sistema de almacenamiento se debe dar aviso y no realizar nada
  - En caso de que la operación sea satisfactoria se debe realizar y avisar por pantalla.
- -g:
  - Si el archivo no existe en la carpeta "archivos originales" se debe dar aviso.
  - Si el archivo ya existe en el sistema de almacenamiento, se debe dar aviso y no realizar nada.
  - En caso de que la operación sea satisfactoria se debe realizar y avisar por pantalla.
- -e:
  - Si el archivo no existe en la carpeta "archivos originales" se debe dar aviso.
  - En caso de que la operación sea satisfactoria se debe realizar y avisar por pantalla.
- -r:

- Si el archivo no existe en la carpeta "archivos originales" se debe dar aviso.
  - Si no se pueden restaurar los archivos faltantes, se debe dar aviso y no realizar la operación.
  - En caso de que la operación sea satisfactoria se debe realizar y avisar por pantalla.
- -R:
    - Si el archivo no existe en la carpeta "archivos originales" se debe dar aviso.
    - Si falta alguna parte en los discos originales y los discos de respaldo se debe dar aviso y no hacer nada.
    - En caso de que la operación sea satisfactoria se debe realizar y avisar por pantalla.
- -m:
    - Si no existe un sistema de almacenamiento creado se debe dar aviso
    - Si existe un sistema de almacenamiento, se deben listar todos los archivos. Los archivos corruptos se deben mostrar con un \* y los archivos que no se puedan restaurar por falta en el original y en el respaldo, se deben mostrar con \*\*.

## 6 Ejemplo

Si se guarda un archivo(A) de tamaño 5500 bytes y se tiene un sistema de almacenamiento con 6 discos, estos deben quedar de la siguiente forma:

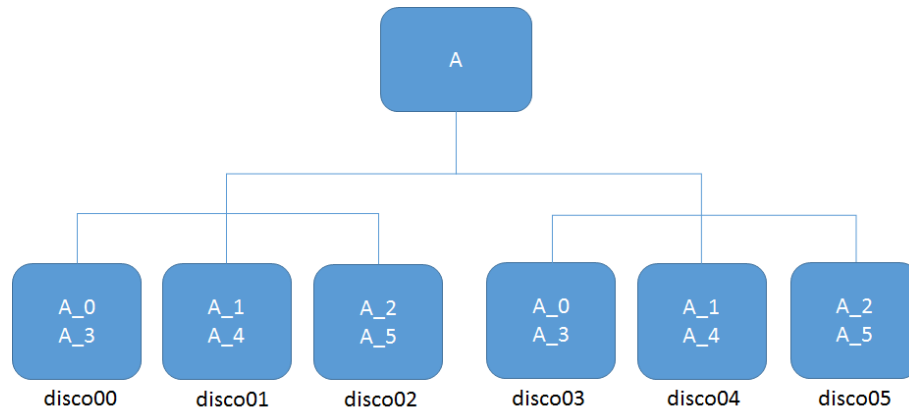


Figura 2: Ejemplo

Disco 00	
├ A 0 .....	1024 bytes
└ A 3 .....	1024 bytes
Disco 01	
├ A 1 .....	1024 bytes
└ A 4 .....	1024 bytes
Disco 03	
├ A 2 .....	1024 bytes
└ A 5 .....	380 bytes
Disco 04	
├ A 0 .....	1024 bytes
└ A 3 .....	1024 bytes
Disco 05	
├ A 1 .....	1024 bytes
└ A 4 .....	1024 bytes
Disco 06	
├ A 2 .....	1024 bytes
└ A 5 .....	380 bytes

Figura 3: Distribución y tamaño de los archivos en discos



## 7 Evaluación

La correcta implementación de la tarea obtendrá nota 7.0, se realizara una pauta de corrección la cual sera dada a conocer a la entrega de las notas. Cualquier situación que no esté contemplada en el enunciado será resuelto el ayudante. El plazo de entrega es el martes 30 de septiembre antes de las 9:30 hrs.