Ejercicio 15.1 Comparación de planificadores de E/S

A continuación proveemos un script que se usa para comparar planificadores de E/S:

```
#!/bin/bash

#/*
# * The code herein is: Copyright the Linux Foundation, 2014
# * Author J. Cooperstein
# *
# * This Copyright is retained for the purpose of protecting free
# * redistribution of source.
# *
# * This code is distributed under Version 2 of the GNU General Public
# * License, which you should have received with the source.
# *
# */

NMAX=8
NMEGS=100
[[ -n $1 ]] && NMAX=$1
[[ -n $2 ]] && NMEGS=$2

echo Doing: $NMAX parallel read/writes on: $NMEGS MB size files

TIMEFORMAT="%R   %U   %S"

#################################################################
# simple test of parallel reads
do_read_test(){
    for n in $(seq 1 $NMAX) ; do
        cat file$n > /dev/null &
    done
# wait for previous jobs to finish
    wait
}

# simple test of parallel writes
do_write_test(){
    for n in $(seq 1 $NMAX) ; do
        [[ -f fileout$n ]] && rm -f fileout$n
        (cp file1 fileout$n && sync) &
    done
# wait for previous jobs to finish
    wait
}

# create some files for reading, ok if they are the same
create_input_files(){
    [[ -f file1 ]] || dd if=/dev/urandom of=file1 bs=1M count=$NMEGS
    for n in $(seq 1 $NMAX) ; do
        [[ -f file$n ]] || cp file1 file$n
    done
}

echo -e "\ncreating as needed random input files"
create_input_files
```



```
#############################################################
# begin the actual work

# do parallel read test
echo -e "\ndoing timings of parallel reads\n"
echo -e " REAL    USER    SYS\n"
for iosched in noop deadline cfq ; do
    echo testing IOSCHED = $iosched
    echo $iosched > /sys/block/sda/queue/scheduler
    cat /sys/block/sda/queue/scheduler
#    echo -e "\nclearing the memory caches\n"
    echo 3 > /proc/sys/vm/drop_caches
    time do_read_test
done
#############################################################
# do parallel write test
echo -e "\ndoing timings of parallel writes\n"
echo -e " REAL    USER    SYS\n"
for iosched in noop deadline cfq ; do
    echo testing IOSCHED = $iosched
    echo $iosched > /sys/block/sda/queue/scheduler
    cat /sys/block/sda/queue/scheduler
    time  do_write_test
done
#############################################################
```

Si está tomando la versión autodidacta en línea de este curso, encontrará el código fuente disponible para su descarga en la pantalla **Lab**.

Recuerde darle permisos de ejecución con:

```
$ chmod +x ioscript.sh
```

Lo que viene a continuación es una explicación del funcionamiento del script y de cómo usarlo:

El script realiza lo siguiente:

- Cambiar entre los planificadores de E/S disponibles en un disco duro, mientras se hace un número configurable de lecturas y escrituras en paralelo de archivos de tamaño ajustable.
- Pruebas de lectura y escritura en pasos por separado.
- Al llevar a cabo las pruebas de lectura, asegúrese de que está leyendo desde el disco y no desde las páginas en memoria caché. Puede forzar que el caché se escriba en disco ejecutando el comando:

  ```
  $ echo 3 > /proc/sys/vm/drop_caches
  ```

  antes de realizar el test de lectura. Puede hacer **cat** a `/dev/null` para evitar escribir en el disco.
- Asegúrese de que todas las lecturas están completas antes de obtener información del tiempo tomado; esto puede llevarse a cabo ejecutando el comando **wait** en una shell.
- Las pruebas de escritura se hacen simplemente copiando un archivo (el cual estará en el caché luego de la primera lectura) múltiples veces de forma simultánea. Para asegurarse que todas las operaciones de escritura se han completado antes de obtener información del tiempo tomado, puede ejecutar el comando **sync**.

El script proporcionado toma dos argumentos. El primero es el número simuntáneo de lecturas y escrituras a realizar. El segundo es el tamaño en MB de cada archivo.

Este script debe ejecutarse como root, dado a que obtiene valores desde los árboles de directorios `/proc` y `/sys`.

Compare los resultados obtenidos usando diferentes planificadores de E/S.

Para realizar una exploración adicional, podría intentar cambiar algunos de los parámetros ajustables y ver cómo varían los resultados.

