

DESARROLLO DE SISTEMAS DISTRIBUIDOS

Mejoras de desempeño con el uso de Hilos

Elaborado por: Ukranio Coronilla

Existen situaciones donde será conveniente la aplicación de hilos, y otras donde no, por tal motivo es conveniente evaluar si un proceso está orientado a uso de CPU, de RAM o de Disco Duro. En la presente práctica haremos pruebas de desempeño para determinar en qué situaciones utilizar hilos en procesos orientados cien por ciento al uso de estos recursos; pero en situaciones reales tendremos que simular el, porcentaje de su uso lo más cercano a nuestra aplicación con ayuda de la función `usleep()`.

Importante: Todas las pruebas se deben realizar en la misma computadora (preferentemente la más potente del equipo de trabajo), sin ejecutar LINUX en máquina virtual. Evite ejecutar simultáneamente otras aplicaciones que puedan afectar sus mediciones.

USO DE DISCO DURO

El siguiente código guarda en un archivo (cuyo nombre se pasa como único parámetro en la línea de comandos), una línea de caracteres solicitados dentro del programa.

```
#include <iostream>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>
#include <cstring>

using namespace std;

char buffer[BUFSIZ];

int main(int argc, char *argv[])
{
    int destino;
    string linea;

    if(argc != 2){
        cout << "Forma de uso: " << argv[0] << " nombre_del_archivo\n";
        exit(0);
    }
    cout << "Introduce texto para guardar en el archivo y ENTER para terminar. Longitud máxima de caracteres = " << BUFSIZ << endl;
    getline(cin, linea);
    strcpy(buffer, linea.c_str());
    //Abre un archivo para escritura, si no existe lo crea, si existe lo trunca, con permisos rw-
    if((destino = open(argv[1], O_WRONLY|O_TRUNC|O_CREAT, 0666)) == -1)
    {
        perror(argv[1]);
        exit(-1);
    }
    write(destino, buffer, strlen(buffer));
    fsync(destino);
    close(destino);
}
```

Ejercicio 1

Analice su funcionamiento y modifíquelo para tomar el tiempo que demora en guardar en un archivo la cadenota de $26*26*26*100$ tokens de tres caracteres que se realizó en la práctica de desempeño con las siguientes consideraciones (genere primero la cadenota y después guárdela en disco, también verifique en cada caso que el archivo de salida contenga 7,030,400 Bytes):

- A) Guarde en operaciones sucesivas de escritura de un byte por un byte.
- B) Guarde en operaciones sucesivas de escritura de un token por un token.
- C) En operaciones sucesivas de escritura de BUFSIZ bytes por BUFSIZ bytes.

¿Cuál es la más conveniente de las tres? ¿Por qué demoran distintos tiempos si la cantidad de información que se almacena es la misma en los tres casos? ¿Para qué se está usando la función `fsync`?

USO DE RAM

Ejercicio 2

El siguiente programa hace uso intensivo de la memoria RAM:

```
#include <iostream>

using namespace std;

#define numeroElementos 100000000

int main(){
    int *arreglo, i;

    arreglo = new int[numeroElementos];

    for(i = 0; i < numeroElementos; i++)
        arreglo[i] = 0;

    for(i = 0; i < 100000000; i++){
        arreglo[rand()%numeroElementos] = rand();
    }
}
```

Modifique el número de ciclos que se ejecuta el segundo for, para que se tarde en ejecutar aproximadamente el mismo tiempo que el programa del inciso A en el ejercicio 1, al cual llamaremos RAM.cpp. Observe con el monitor de sistema que se reserva la memoria, y en el código que se accede continuamente a la RAM.

USO DE CPU

Ejercicio 3

Retome el programa de uso de CPU con operaciones matemáticas que se realizó en la práctica de desempeño, e inicialice el valor de Max de manera que su programa tarde en ejecutarse aproximadamente lo mismo que el inciso A del ejercicio1, al cual llamaremos CPU.cpp.

USO DE HILOS

Ejercicio 4

Ahora programe los casos siguientes con hilos concurrentes:

- 1) En un hilo ejecute el programa del inciso A en el ejercicio 1 al cual llamaremos DD.cpp, y en otro hilo ejecute el mismo programa pero que almacene la cadenota en un archivo distinto.
- 2) En un hilo ejecute el programa RAM y en otro hilo el mismo programa
- 3) En un hilo ejecute el programa CPU y en otro hilo el mismo programa.
- 4) En un hilo ejecute el programa DD y en otro hilo ejecute el programa CPU
- 5) En un hilo ejecute el programa DD y en otro hilo ejecute el programa RAM
- 6) En un hilo ejecute el programa CPU y en otro hilo ejecute el programa RAM
- 7) En un hilo ejecute el programa CPU, en otro hilo ejecute el programa DD, y en otro hilo ejecute el programa RAM.

Observe el monitor de sistema cuando ejecute los programas y mida los tiempos de ejecución. Compare los tiempos obtenidos en este ejercicio, con los que se obtuvieron en los ejercicios anteriores, y analice la información para concluir para este caso en qué situaciones es conveniente el uso de hilos, en cuáles no, y cuáles son las combinaciones más convenientes. Tenga esto presente siempre que vaya a programar hilos en sus aplicaciones distribuidas.