

I OBJETIVOS

- Medir y comparar la ejecución de procesos que requieren uso intensivo del CPU y procesos que requieren E/S.

II BIBLIOGRAFÍA

- Silberschatz & Galving, “SISTEMAS OPERATIVOS”, McGraw Hill.

III RECURSOS

- Una PC con Linux
 - Las herramientas de desarrollo bajo Linux
- Se recomienda **no usar máquinas virtuales**, para este caso sería preferible usar Cygwin si no se dispone de una computadora con Linux

IV ACTIVIDADES

1 *Ejecución de procesos concurrentes que requieren uso intensivo del CPU y E/S*

1.1 **Compilación y ejecución del programa de prueba**

Copia el programa que se muestra en la Figura 1 a un archivo de texto llamado `cpuyes.c`. Este programa de prueba realiza un ciclo que ejecuta instrucciones del CPU y después permanece bloqueado durante un tiempo tal que la duración final del proceso es de 10 segundos. Este proceso recibe como argumento el porcentaje de tiempo que hará uso intensivo del CPU. Por ejemplo:

```
$ ./cpuyes 100
```

100 % de su tiempo hace uso del CPU

```
$ ./cpuyes 50
```

50 % de su tiempo hace uso del CPU y el 50 % restante entrada y salida

```
$ ./cpuyes 0
```

100 % de su tiempo hace uso de E/S

Para que la duración del programa `cpuyes` sea de 10 segundos es necesario ajustar el valor de la constante `CICLOS`. Realiza la modificación y prueba ejecutando un proceso que demande 100 % de su tiempo el CPU de manera que la ejecución del programa se haga en 10 segundos.

Compila el programa utilizando el siguiente comando:

```
$ gcc -o cpuyes cpuyes.c
```

El archivo `cpuyes` es el programa ejecutable. Para ejecutarlo solo teclea:

```
$ ./cpuyes 100
```

```
#include <stdio.h>
#include <sys/time.h>

#define CICLOS 30000000 // Modifique este valor para que la duración sea 10
segundos

int main(int argc, char *argv[])
{
    long long start_ts;
    long long stop_ts;
    long long elapsed_time;
    long lElapsedTime;
    struct timeval ts;

    int i, j;
    int porc;

    if(argc<2)
        porc=100;
    else
        porc=atoi(argv[1]);

    printf("CPU=%d, E/S=%d\n", porc, 100-porc);

    gettimeofday(&ts, NULL);
    start_ts = ts.tv_sec * 1000000 + ts.tv_usec; // Tiempo inicial
```

```

for (i=0; i<porc ; i++)      // Ejecutar instrucciones del CPU
    for(j=0;j<CICLOS;j++); // Uso intensivo del CPU

usleep((100-porc)*100000);

gettimeofday(&ts, NULL);
stop_ts = ts.tv_sec * 1000000 + ts.tv_usec; // Tiempo final

elapsed_time = stop_ts - start_ts;
printf("proceso %d, %d microsegundos\n",getpid(),elapsed_time);
}

```

Figura 1. Programa de prueba que ejecuta instrucciones haciendo uso del CPU y E/S

1.2 Arrancador de procesos

El programa arranca que se muestra en la Figura 2 permite indicar cuantos procesos cpuyes se ejecutarán concurrentemente y que porcentaje de su tiempo demandarán CPU. Al final de la ejecución mostrará la tasa de trabajos¹.

Para compilar solo teclee:

```
$ gcc -o arranca arranca.c
```

```

#include <stdio.h>
#include <sys/time.h>

int main()
{
    long long start_ts;
    long long stop_ts;
    long long elapsed_time;
    long lElapsedTime;
    struct timeval ts;
    int pid;
    int i,procs,pcpu;
    char spcpu[5];
    int status;
    float tasa;

    printf("Procesos ..... :");
    scanf("%d",&procs);

```

¹ Tasa de trabajos son los trabajos terminados en una unidad de tiempo que es un segundo.

```
printf("Porcentaje uso de CPU :");
scanf("%d",&pcpu);

sprintf(spcpu,"%d",pcpu);

gettimeofday(&ts, NULL);
start_ts = ts.tv_sec; // Tiempo inicial

for(i=0;i<procs;i++)
{
    pid=fork();
    if(pid==0)
        execlp("nice","nice","--adjustment=0","./cpuyes",spcpu,0);
}

for(i=0;i<procs;i++)
    wait(&status);

gettimeofday(&ts, NULL);
stop_ts = ts.tv_sec; // Tiempo final

elapsed_time = stop_ts - start_ts;
printf("-----\n");
printf("TIEMPO TOTAL, %d segundos\n",elapsed_time);
tasa=(float) procs / (float) elapsed_time;
printf("Tasa de trabajos = %1.4f\n",tasa);
}
```

Figura 2.- Programa arranca.c

2 Ejecución de procesos que demandan uso intensivo del CPU

Ejecuta el programa arranca solicitando 1 proceso que demande 100 % de su tiempo CPU. Toma nota de la tasa de trabajos.

Repite el experimento con 2,3,4, hasta llegar a 10 procesos que demanden 100% CPU

Si el procesador tiene tecnología HT o es Multicore, realiza los experimentos deshabilitando procesadores de manera que solo uno lo ejecute. Esto se puede hacer utilizando el programa taskset, ejemplo:

```
$ taskset 01 ./arranca
```

2.1 Resultados

Como se muestra en la Figura 3, realice una gráfica. Indique cuantos procesadores ejecutaron el proceso.

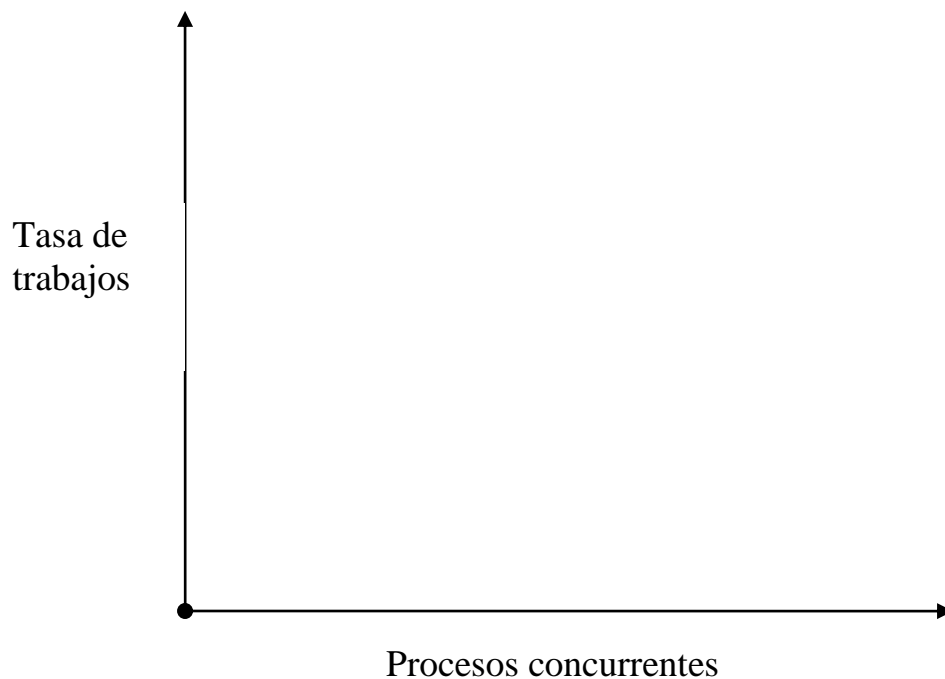


Figura 3. Gráfica de tiempo con respecto a los procesos en ejecución.

2.2 Preguntas

- 1.- Según la gráfica desarrollada, ¿mejora la tasa de trabajos agregando procesos?
- 2.- Explique por qué
- 3.- En la grafica desarrollada ¿Dónde se da una asíntota?
- 4.- Explique el por qué se da una asíntota en ese punto.

3 *Ejecución de procesos concurrentes que permanecen bloqueados (puede ser en espera de E/S)*

Ejecuta el programa `arranca` solicitando 1 proceso que demande 0 % de su tiempo CPU, es decir 100% de su tiempo demanda entrada y salida. Toma nota de la tasa de trabajos.

Repite el experimento con 2,3,4, hasta llegar a 10 procesos.

3.1 Preguntas

- 1.- Según la gráfica desarrollada, ¿mejora la tasa de trabajos agregando procesos?
- 2.- Explique por qué
- 3.- En la grafica desarrollada ¿Dónde se da una asíntota?
- 4.- Explique el por qué se da una asíntota en ese punto.

4 *Ejecución de procesos concurrentes que el 50 % de su tiempo utilizan CPU y el 50 % de su tiempo permanecen bloqueados (puede ser en espera de E/S)*

Ejecuta el programa `arranca` solicitando 1 proceso que demande 50 % de su tiempo CPU, es decir el 50% de su tiempo restante demanda entrada y salida. Toma nota de la tasa de trabajos.

Repite el experimento con 2,3,4, hasta llegar a 10 procesos.

Como se muestra en la Figura 3, realice una gráfica. Indique las características del CPU (si es Core Solo, HT, Core Duo, Quad Core).

4.1 Preguntas

- 1.- Según la gráfica desarrollada, ¿mejora la tasa de trabajos agregando procesos?
- 2.- Explique por qué
- 3.- En la grafica desarrollada ¿Dónde se da una asíntota?

- 4.- Explique el por qué se da una asíntota en ese punto.
- 5.- Considerando que se tiene el dato del porcentaje de tiempo que los trabajos usan CPU, y este porcentaje puede ser cualquier valor. Determine una fórmula en función del porcentaje de tiempo de CPU donde pueda calcularse el número máximo del total de trabajos por segundo que puede obtenerse si agregamos un número infinito de procesos.

V ENTREGA

Esta práctica se realiza de manera individual²

Subir el reporte en **formato PDF** en el apartado correspondiente en Moodle a más tardar el martes 12 de Febrero a más tardar a las 11:59 PM.

Traer el **reporte impreso** a clase del miércoles 13 de Febrero para discutir y concluir sus resultados con el grupo.

² Al ser trabajo individual se entiende que no se permite ningún tipo de colaboración.