

Practica 5: Multiplicación de matrices con procesos

Alumno(s):

Hernández Pérez Priscila Itzel
Méndez Gutiérrez Néstor Javier
Franco Tenorio Roberto Carlos

Profesor – M.I. Marcela Ortiz Hernández

22 de septiembre de 2019

Semestre - 2019-2020/I

Multiplicación de matrices con procesos.

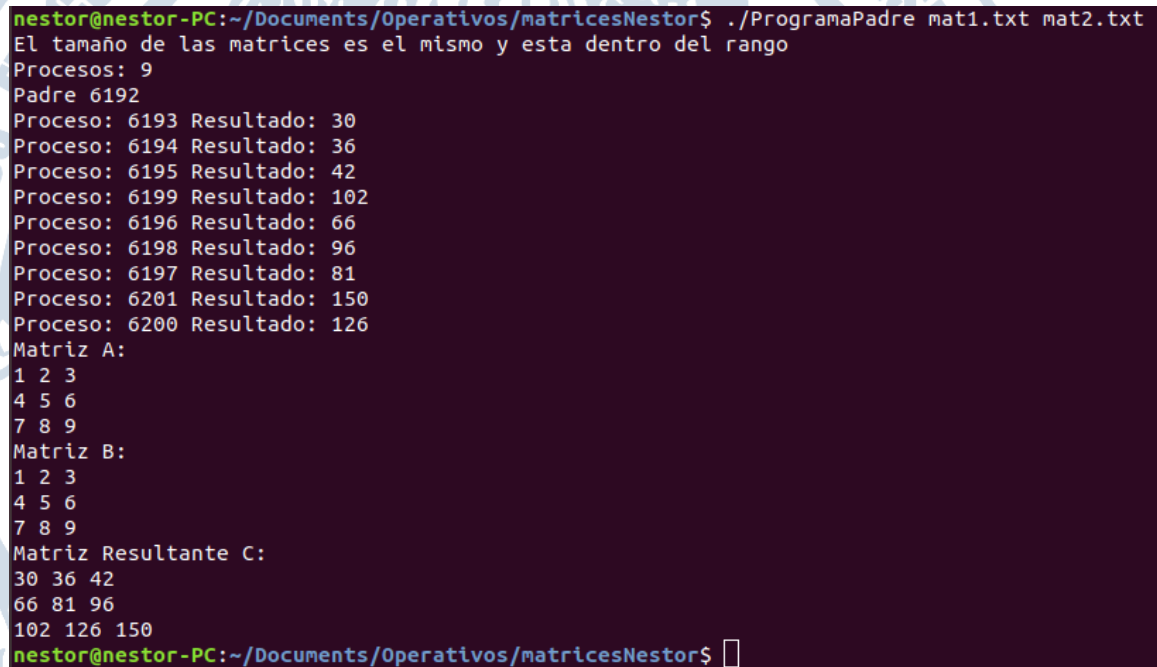
1. Introducción

El objetivo de esta practica es implementar un programa que realiza una multiplicación de dos matrices de $N \times N$ para la cual se han de usar la funciones `fork()`, `execv()` y `wait()` con las cuales se han de crear y manipular N procesos hijos para calcular cada uno de los elementos de la matriz resultante.

2. Datos

“no aplica”

3. Figuras, imágenes o gráficos



```
nestor@nestor-PC:~/Documents/Operativos/matricesNestor$ ./ProgramaPadre mat1.txt mat2.txt
El tamaño de las matrices es el mismo y esta dentro del rango
Procesos: 9
Padre 6192
Proceso: 6193 Resultado: 30
Proceso: 6194 Resultado: 36
Proceso: 6195 Resultado: 42
Proceso: 6199 Resultado: 102
Proceso: 6196 Resultado: 66
Proceso: 6198 Resultado: 96
Proceso: 6197 Resultado: 81
Proceso: 6201 Resultado: 150
Proceso: 6200 Resultado: 126
Matriz A:
1 2 3
4 5 6
7 8 9
Matriz B:
1 2 3
4 5 6
7 8 9
Matriz Resultante C:
30 36 42
66 81 96
102 126 150
nestor@nestor-PC:~/Documents/Operativos/matricesNestor$
```

4. Preguntas

1. Escriba el algoritmo del proceso padre, explicando como crea los procesos necesarios, como espera los resultados y como imprime el resultado de la multiplicación de las matrices.
 - Se iguala la variable `pid` a la función `fork()` la cual crea el primer proceso hijo.
 - Se realizan dos ciclos **for** anidados cada uno se repite N veces, donde N es el tamaño de las matrices las cuales deben ser matrices cuadradas.
 - En el primer ciclo se convierte el iterador `i` a cadena y en segundo ciclo se obtiene el iterador `j` y de igual forma se convierte a cadena.
 - Se verifica el valor del `pid` donde:
 - Si el valor es negativo sabemos que el proceso no se pudo crear.

- Si el valor es cero, el proceso se creó y se está ejecutando, en este caso creamos la cadena de caracteres que será parámetro del proceso hijo, esta cadena contiene en la posición **cero** el nombre del ejecutable, en la posición **uno** y **dos**, se guardan las posiciones del renglón y la columna que el proceso hijo tiene que multiplicar y por último en las posiciones **tres** y **cuatro** se guardan los nombres de los archivos que contienen las matrices.
 - Si es positivo indica que el proceso padre se está ejecutando. En este caso se obtiene el PID del proceso hijo que se acaba de crear y se almacena temporalmente en la matriz resultante.
 - Después de creados los procesos hijos se realiza un ciclo **for** el cual se repetirá tantas veces como el número de procesos hijos creados, en este ciclo se iguala una variable de tipo `pid_t` a la función `wait()`, y se le pasa como parámetro un apuntador a una variable entera, el valor modificado de la variable que usó `wait` se pasa como parámetro para la macro `WEXITSTATUS` mediante la cual obtenemos el valor que regresa el proceso hijo que acaba de terminar.
2. Explique el uso de las funciones `fork()`, `execv()` y `wait()`.
- Fork: esta función crea la bifurcación entre el proceso padre e hijo.
Execv: ejecuta el código del proceso hijo.
Wait: función que nos ayuda a esperar a que un proceso hijo termine su ejecución.
3. Explique el formato que utilizó en los archivos que contienen las matrices.
- El primer renglón del archivo de texto contiene un número el cual indica la cantidad de renglones y columnas de la matriz, después del primer renglón se encuentra la matriz de números los cuales deben de ser iguales o mayores a cero y menores o iguales a 9.
4. Escriba cual es la orden en la línea de comandos para ejecutar su práctica. Indique y explique cada uno de sus argumentos.
- `./ProgramaPadre mat1.txt mat2.txt`
- a. Primer parámetro: nombre del ejecutable
 - b. Segundo parámetro: nombre del archivo que contiene la matriz A.
 - c. Tercer parámetro: nombre del archivo que contiene la matriz B.
5. En este punto del curso, esta práctica se crea explícitamente un proceso que colabora con otros procesos. Enumere al menos un beneficio y un inconveniente de esta metodología de programación.

Un beneficio obvio es que algunos programas se pueden paralelizar y de esta manera varios procesos realizan cálculos al mismo tiempo.

Un inconveniente es que no es sencillo comprender como funcionan estos programas que se ejecutan de forma paralela y también es que es mas complejo mantener el control de los datos que manipulan los distintos procesos.

6. ¿Cuál fue la parte más difícil de esta práctica?

La parte más compleja fue la comprensión de como el programa debía de funcionar y también la manipulación de los procesos una vez que estos eran creados, ya que no siempre estos procesos terminaban antes que el padre, esto a pesar de que se ejecutaba la función wait().

¿Faltaban habilidades de programación?

Creo que las habilidades son las requeridas para la solución del problema.

Proporcione un comentario honesto para que podamos usarlo para ajustar el alcance de la práctica para futuros cursos de Sistemas Operativos.

Creo que quizás se debería explicar el funcionamiento de procesos con un programa mas complejo y que se profundice en el funcionamiento de este programa.

5. Conclusiones

En esta práctica se aplicaron los conocimientos obtenidos en clase acerca del funcionamiento de los procesos, como es que se crean, ejecutan y como obtener un resultado de estos, fue una práctica un tanto compleja mas no imposible, y nos ayudo bastante en la comprensión de el funcionamiento de procesos.