

# Práctica: Memoria compartida

Se desea implementar un conjunto de procesos que permitan la simulación de una calculadora binaria con las 4 operaciones básicas.

La implementación se realizará mediante el uso de procesos independientes, para ello se invocarán las llamadas al sistema `fork` y `exec`.

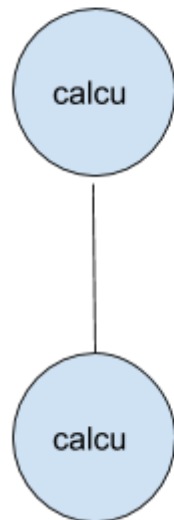
## Especificaciones.

- La calculadora a implementar tendrá las 4 operaciones básicas: suma (+), resta(-), multiplicación(\*) y división (/).
- La calculadora además implementará un nuevo operador #, el cual significa que se deben realizar las 4 operaciones básicas.
- La calculadora deberá realizar las operaciones utilizando un proceso independiente a él, por lo que si se desea solo hacer una operación se creará un solo proceso independiente, pero si se desean hacer las 4 operaciones, deberán crearse 4 procesos independientes, uno por cada operación.

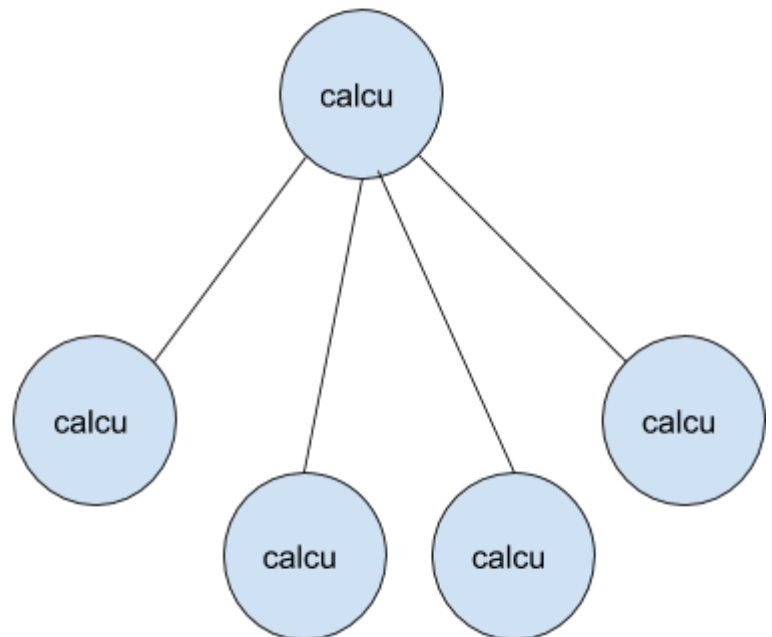
## Detalle técnicos.

- El programa recibirá mediante consola la expresión binaria a resolver, los cuales son:
  - operador1+operador2
  - operador1-operador2
  - operador1\*operador2
  - operador1/operador2
  - operador1#operador2
- Los operandos pueden ser reales o enteros.
- En caso de no encontrar uno de los 5 operadores reconocidos, el programa terminará, notificando el error ocurrido.
- El programa lanzará de forma automática el/los proceso(s) independientes, de acuerdo al operador encontrado en la expresión binaria.
- El programa que pone en funcionamiento la calculadora lo llamaremos **“calcu”**, y recibirá como argumentos desde consola la expresión binaria a resolver como una única cadena.
- Una vez que **“calcu”** recibe la cadena que contiene la expresión binaria, debe analizarla y tratarla para obtener el operador y los operandos los cuales deben ser casteados como un número.
- Una vez que determina cuál es la(s) operación(es) a realizar, pueden hacerse solo 1 o las 4 operaciones, el programa **“calcu”** se duplicará usando la llamada al sistema fork, como se muestra en la figura:

**Si se elige solo 1 operación**

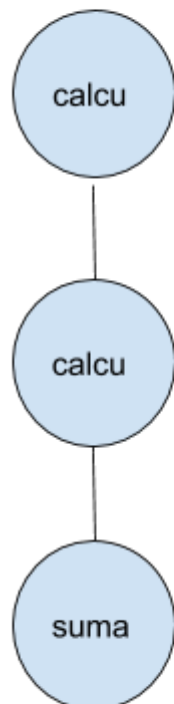


**Si se elige las 4 operaciones**

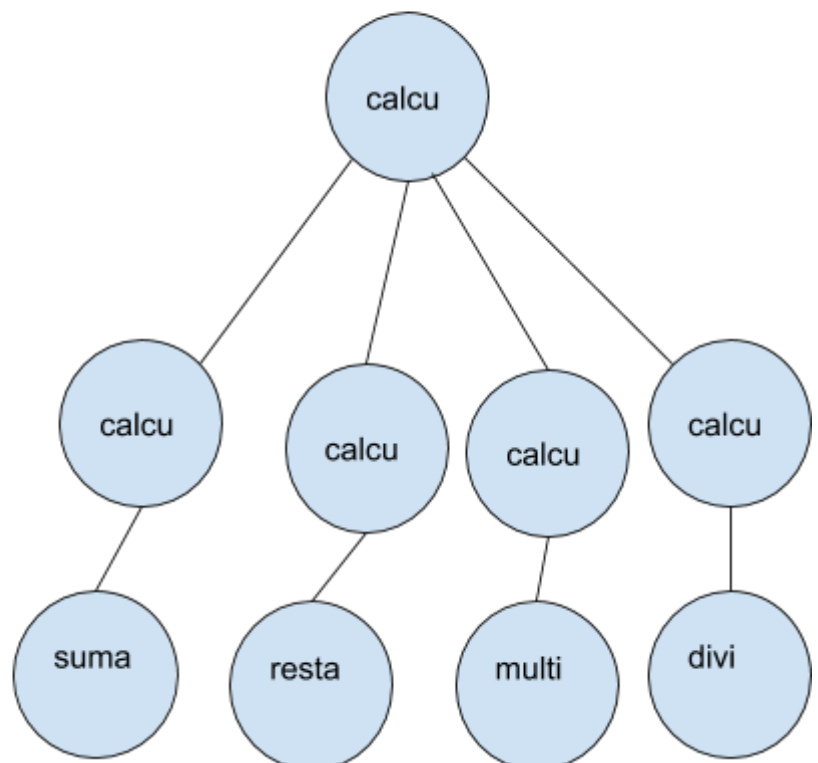


- Los procesos duplicados, es decir los procesos hijos, serán reemplazados por el proceso experto en resolver una operación, usando la llamada al sistema exec, como se muestra en la figura.

**Si se elige solo 1 operación**



**Si se elige las 4 operaciones**



- El proceso padre “**calcu**” se comunicará con los procesos expertos mediante memoria compartida, una memoria para cada uno de los procesos expertos.

Dicha memoria contendrá los siguientes campos:

- Operador1
  - Operador2
  - Resultado
  - MensajeError
- 
- Considerando que a través de la llamada al sistema exec, los procesos hijos se pueden comunicar con los procesos que los reemplazarán mediante los argumentos, entonces será a través de ellos, que se les comunicarán el nombre del archivo y el valor entero que se requiere para crear la clave usando la función **ftok**.
  - El proceso padre **“calcu”** tomará los operandos en la memoria compartida, en la memoria con la que hay comunicación entre **“calcu”** y el proceso experto, así el proceso experto obtendrá los operadores de la memoria.
  - Los procesos expertos realizarán la operación y el resultado lo dejará en la memoria compartida, si no existe ningún error, en el caso contrario lo informará mediante un mensaje que también dejará en la memoria compartida. De esta manera el proceso **“calcu”** padre podrá acceder al resultado o error, para así mostrarlos al usuario.

Además se les pide que investiguen cómo obtener mediante comandos la siguiente información y crear un script que lo haga.

- Número máximo de segmentos de memoria compartida que permite crear el sistema.
- Número máximo de segmentos de memoria compartida que puede crear un proceso.
- Tamaño máximo (bytes) que puede tener un segmento de memoria compartida.
- Obtener el tamaño de la memoria compartida X.
- Proceso que crea el segmento de memoria compartida X.
- Procesos que utilizan la memoria compartida X.