Universidad de Costa Rica
Facultad de Ingeniería
Escuela de Ciencias de la Computación e Informática
CI1323 Arquitectura de Computadores

Grupo 1
Prof. Ileana Alpízar Arias

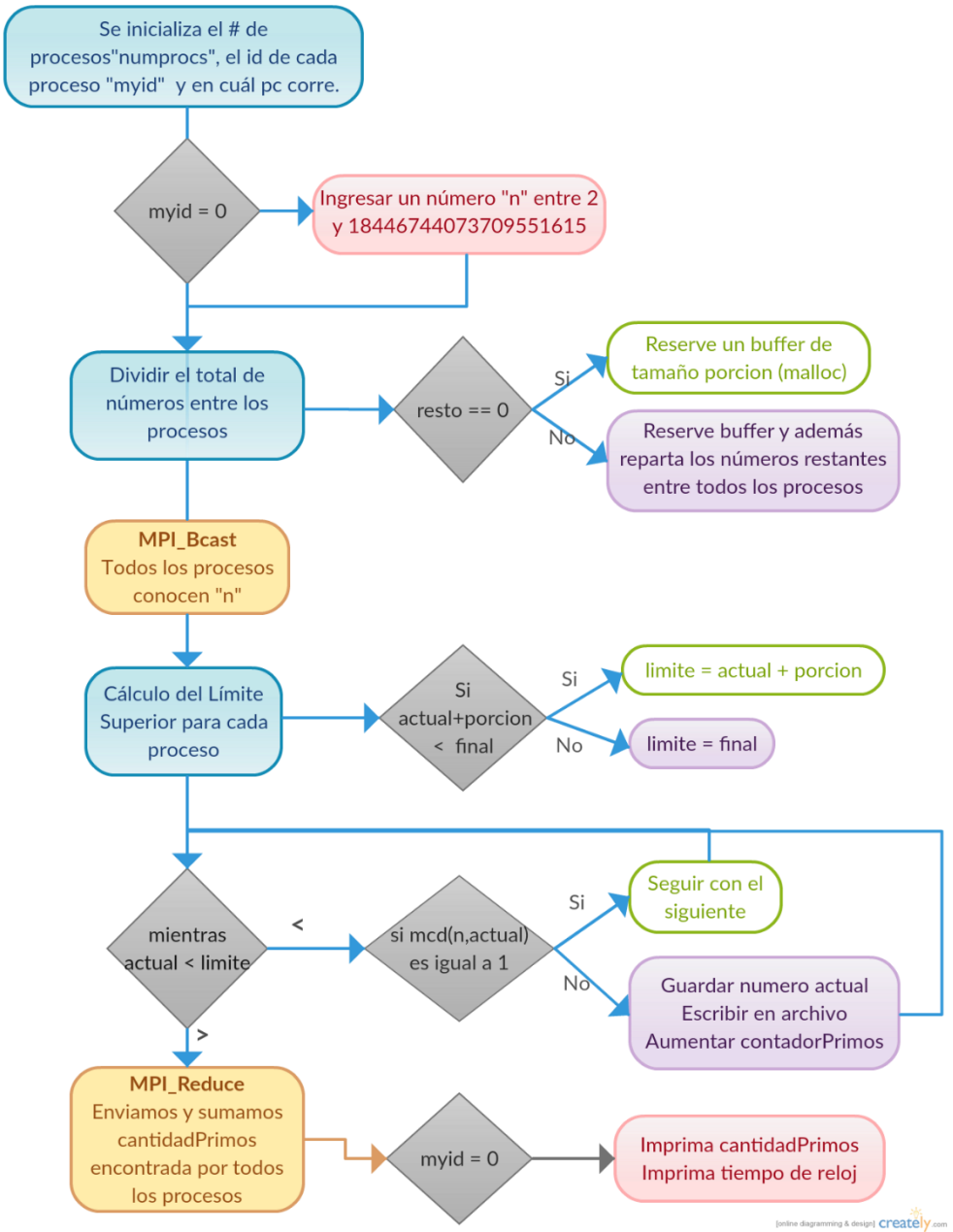
Proyecto: Tarea Programada I “Primos Relativos”.

Estudiantes:
Michelle Cersosimo Morales B21684
Carlos Sanabria Sandoval A75952

Fecha: 4 de septiembre del 2015

Descripción de la lógica completa del programa

Sea:
inicio: 2^27
final: 2^32
total: 2^32 -2^27 +1
porción: total/numprocs
resto = total%numprocs
contadorPrimos = 0
actual = inicio + myid * porción



Algoritmo utilizado para la búsqueda de los primos relativos de un número.

Por definición, los números primos entre sí, no tienen otro divisor más que 1. Por esa razón, se utilizó el algoritmo de Euclides para encontrar el máximo común divisor entre dos números A y B, en donde se siguen los siguientes principios:

- Si B es igual a 0, el máximo común divisor es A y el algoritmo termina.
- En otro caso, el máximo común divisor vuelve a realizarse pero esta vez entre B y A%B.

Para nuestros fines, si el máximo común divisor es igual a 1, significa que entre A y B el único en común que puede dividirlos es el 1, siendo así primos relativos.

```
int mcd(int a, int b) {
        int part, aux;
        part = a%b;
        while (part != 0) {
                aux = b;
                b = part;
                part = aux%b;
        }
        return b;
}
```

Algoritmo *mcd*, Calcula el máximo común divisor entre 2 números.

Modo de crear los sub-rangos

Para definir los subrangos se utiliza el total de números y el número de procesos disponibles. Primero se divide el número total de números posibles a dividir en todos los procesos entre los procesos disponibles. En nuestro caso el rango va desde 2^{27} a 2^{32} o sea entre 134 217 728 y 4 294 967 296 y por lo tanto el total de números posibles es: 4 160 749 569. Por ejemplo si tenemos en total 3 procesos disponibles cada proceso analizaría 1386916523 números en total.

Al ser posible que la división no sea exacta, se toman el resto de la anterior división y se divide nuevamente entre los procesos existentes. A pesar de que nuestro algoritmo considera este escenario se recomienda que el número 4 160 749 569 sea múltiplo del número de procesos para repartir los sub-rangos de manera equitativa.

Problemas no resueltos

A lo largo del desarrollo de la tarea solo se dejó un problema sin resolver el cual consiste en presentar todos los primos relativos en un único archivo, en la actual implementación cada proceso escribe su propio archivo