

Concorrenca e Paralelismo. Bloque II Paralelismo

Práctica 1: Cuenta las apariciones de una letra

Primavera 2023



Cuenta las apariciones de una letra

Código secuencial

```
int i, n, count=0;
char *cadena;
char L;

n = atoi(argv[1]);
L = *argv[2];

cadena = (char *) malloc(n*sizeof(char));
inicializaCadena(cadena, n);

for(i=0; i<n; i++){
    if(cadena[i] == L){
        count++;
    }
}

printf("El numero de apariciones de la letra %c es %d\n", L, count);
free(cadena);
```



Cuenta las apariciones de una letra

Paralelización

- Implementación SPMD *Single Process Multiple Data*
- La E/S (`argv[]/printf`) la hace el proceso 0
- Distribuir n y L a todos los procesos (con `Send/Recv`)
- Reparto de la carga de trabajo en el bucle `for` con “paso” $i += \text{numprocs}$ en lugar de $i++$
- Recoger el número de apariciones por cada proceso (con `Send/Recv`)

Condiciones de realización

- Puntuación: 0.25
- Deadline: 17-21 Abril
- Realización en parejas
- Defensa en laboratorio de prácticas

Cuenta las apariciones de una letra

Formación de las parejas

- Preferiblemente deben ser del mismo grupo de prácticas
 - La defensa se realizará **obligatoriamente** en su grupo de prácticas
- En casos excepcionales se permitirán parejas donde cada miembro esté asignado a un grupo de prácticas distinto
 - La defensa se realizará **obligatoriamente** en alguno de sus grupos de prácticas
 - **Obligatorio** enviar un email a Jorge González Domínguez (jgonzalezd@udc.es) informando de los componentes del grupo. Se responderá a dicho email indicando en qué grupo deberá realizarse la defensa.
 - Ninguna persona que no haya informado vía email podrá defender sus prácticas en un grupo distinto al que esté asignada