

Informe

Diferencias entre Buzy Waiting y Mutex

Alumno: Sebastian Wilde Alarcón Arenas

Algoritmo de para calcular el número pi con Buzy Waiting:

```
void* thread_sum(void*rank)
{
    long my_rank=(long) rank;
    double factor;
    double my_sum=0.0;
    long long i;
    long long my_n=n/thread_count;
    long long my_first_i = my_n*my_rank;
    long long my_last_i=my_first_i+my_n;
    if (my_first_i%2==0) factor=1.0;
    else factor=-1.0;
    for (i=my_first_i;i<my_last_i;i++,factor=-factor)
        my_sum+=factor/(2*i+1);
    while (flag!=my_rank);
    printf("Hello from thread %ld of %d \n",my_rank,thread_count);
    printf("Mi suma %7g\n",my_sum );
    sum+=my_sum;
    flag=(flag+1)%thread_count;
    return NULL;
}
```

Algoritmo de para calcular el número pi con Mutex:

```
void* thread_sum(void*rank)
{
    long my_rank=(long) rank;
    double factor;
    double my_sum=0.0;
    long long i;
    long long my_n=n/thread_count;
    long long my_first_i = my_n*my_rank;
    long long my_last_i=my_first_i+my_n;
    if (my_first_i%2==0) factor=1.0;
    else factor=-1.0;
    for (i=my_first_i;i<my_last_i;i++,factor=-factor)
        my_sum+=factor/(2*i+1);
    printf("Hello from thread %ld of %d suma aqui es %7g\n",my_rank,thread_count,my_sum);
    pthread_mutex_lock(&mutex);
    sum+=my_sum;
    pthread_mutex_unlock(&mutex);
    return NULL;
}
```

Análisis: Entre ambas la más eficiente es Mutex, ya que no hay riesgo que el sistema operativo evite que se cambie los flags y que se quede en un bucle infinito; aparte de ello hay un tiempo muerto por el while además de que en Buzy Waiting empieza desde el primer thread obligándolo a volverse secuencial. Por lo contrario mutex no evita el paralelismo salvo al momento de actualizar la variable.