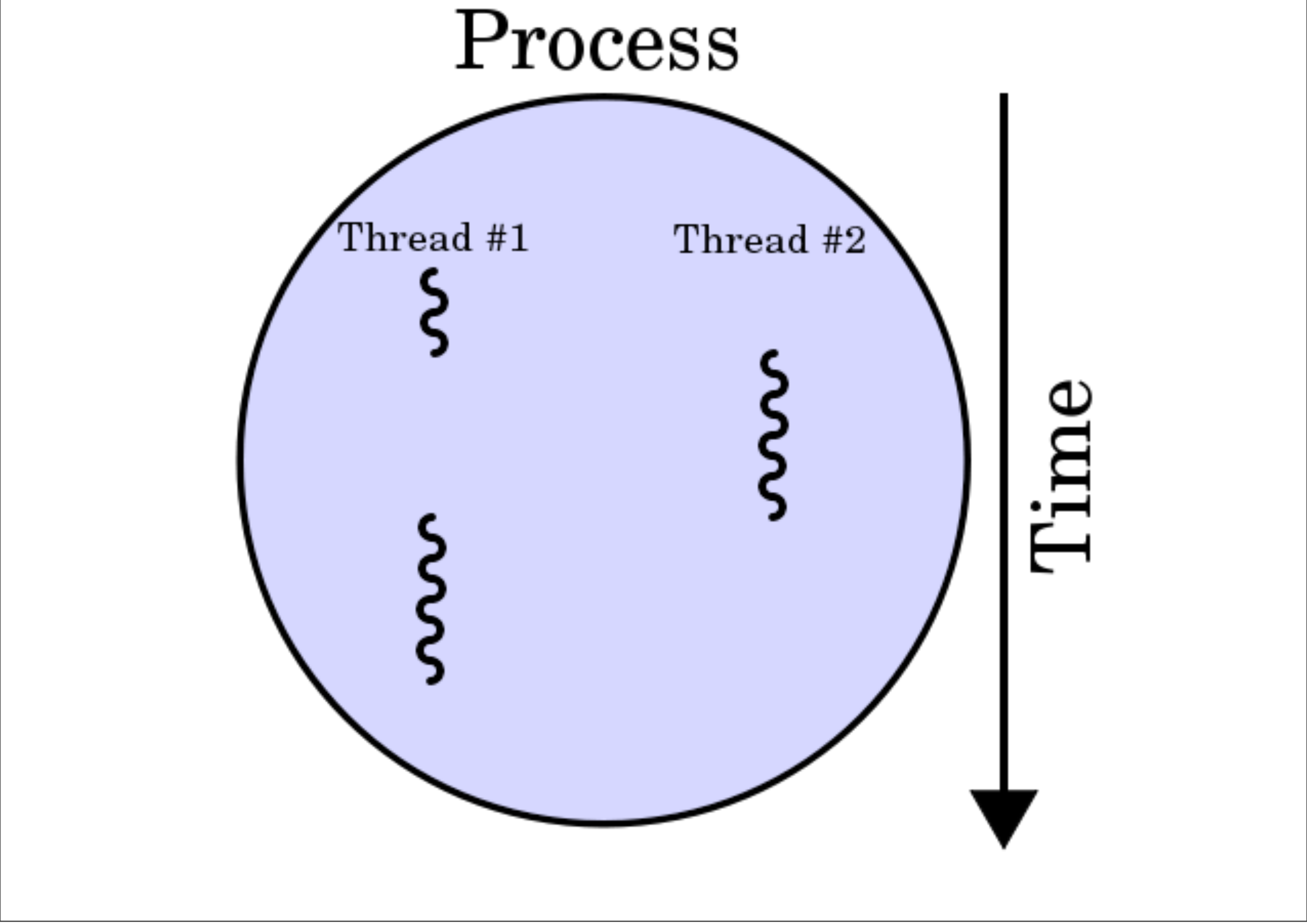
Parallel Programming Models

OpenMP


Process
Thread #1
Thread #2
Time

OpenMP by Example (for)

```
#define SIZE 16000000
```

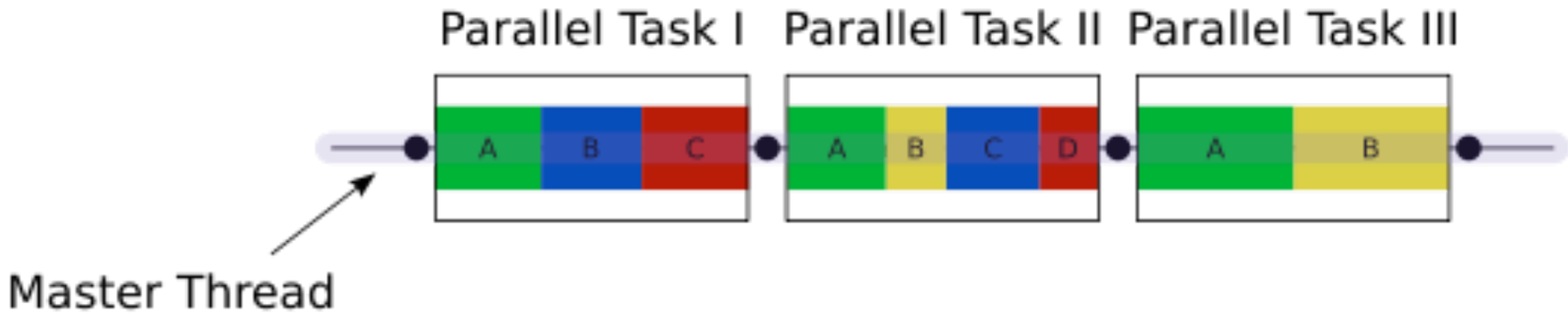
```
for (i = 0; i < SIZE; i++) {
    a[i] = 2 * b[i] + sqrt(c[i]);
}
```

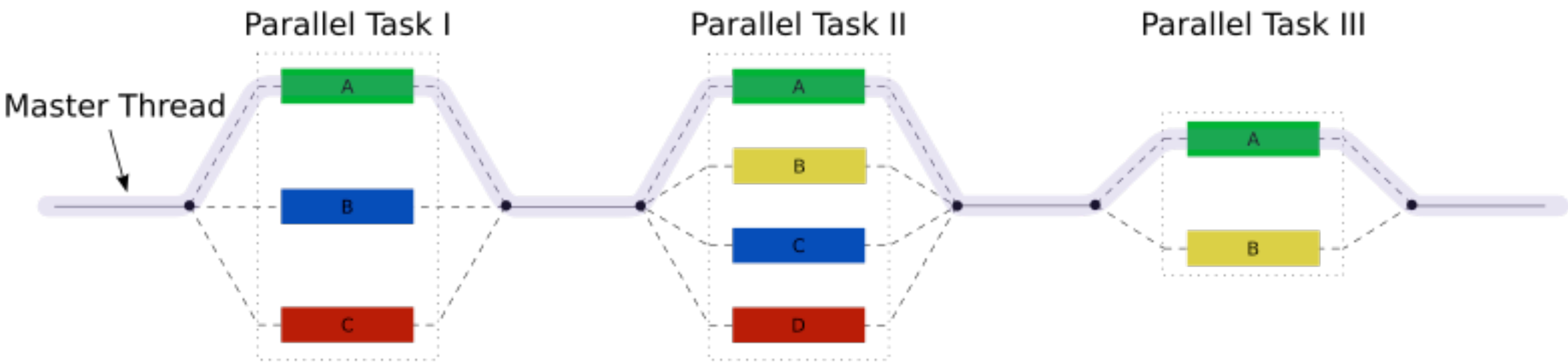
OpenMP by Example (for)

```
#define SIZE 16000000

#pragma omp parallel for
for (i = 0; i < SIZE; i++) {
    a[i] = 2 * b[i] + sqrt(c[i]);
}
```

Fork-Join





Source: wikipedia

OpenMP by Example (for)

```
#define SIZE 16000000

#pragma omp parallel for default(shared)
private(i)
for (i = 0; i < SIZE; i++) {
    a[i] = 2 * b[i] + sqrt(c[i]);
}
```

OpenMP Basic Clause

```
#pragma omp directive clause
STRUCTURED BLOCK
```

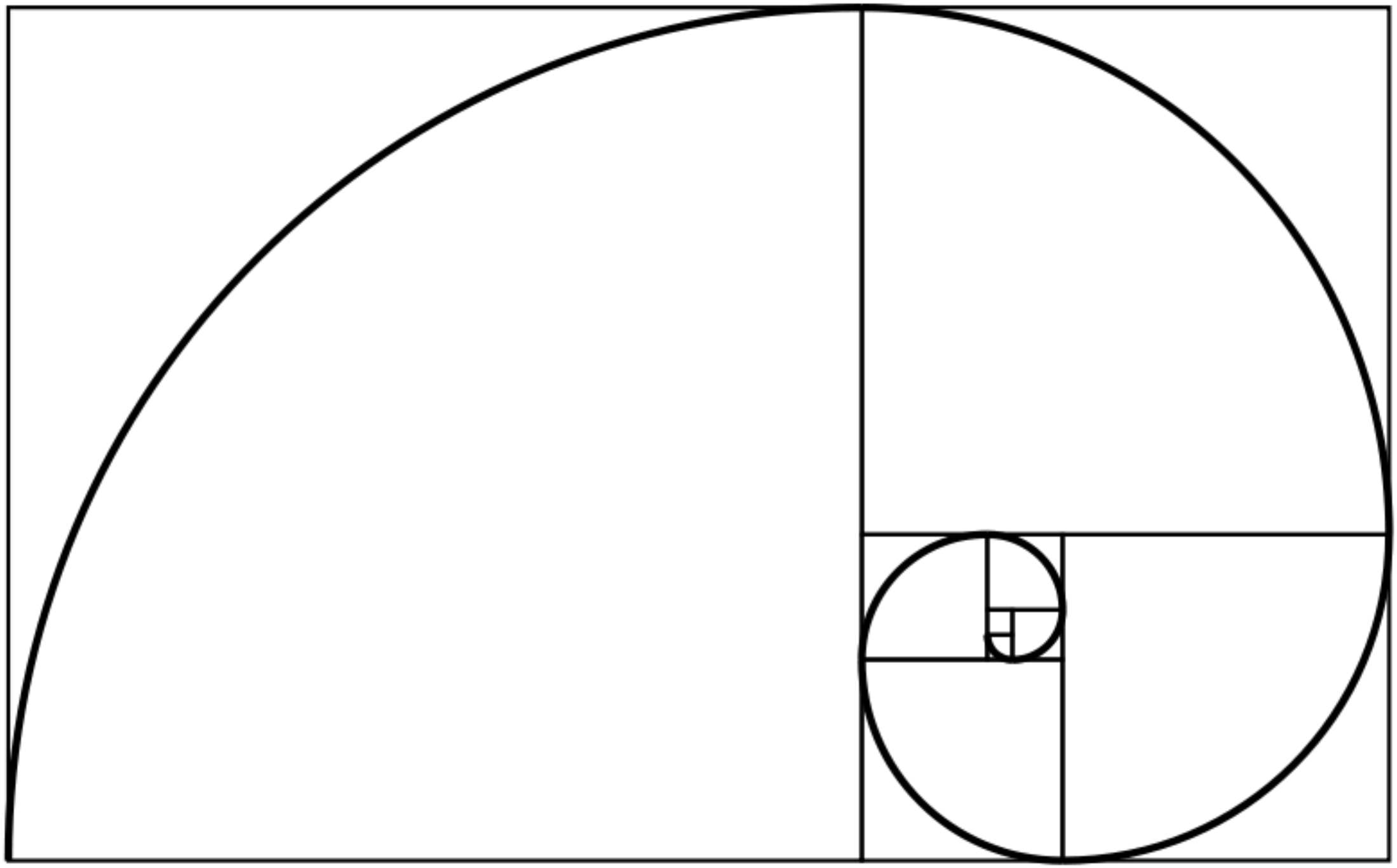
OpenMP Parallel

```
#pragma omp parallel
{
   printf("hello world\n");
}
```

OpenMP Parallel

```
#pragma omp parallel
{
   printf("hello world\n");

   #pragma omp single
   printf("hello again\n");
}
```



Source: wikipedia

OpenMP by Example (tasks)

```
int fib(int n)
{
    if (n < 2) {
        return n;
    }
    else {
        return fib(n - 1) + fib(n - 2);
    }
}
```

OpenMP by Example (tasks)

```
int fib(int n)
{
    if (n < 2) {
        return n;
    }
    else {
        int x, y;
        x = fib(n - 1);
        y = fib(n - 2);
        return x + y;
    }
}
```

OpenMP by Example (tasks)

```
int fib(int n)
{
    if (n < 2) {
        return n;
    }
    else {
        int x, y;
        #pragma omp task default(shared)
        x = fib(n - 1);
        #pragma omp task default(shared)
        y = fib(n - 2);
        return x + y;
    }
}
```

OpenMP by Example (tasks)

```
int fib(int n)
{
    if (n < 2) {
        return n;
    }
    else {
        int x, y;
        #pragma omp task default(shared)
        x = fib(n - 1);
        #pragma omp task default(shared)
        y = fib(n - 2);
        #pragma omp taskwait
        return x + y;
    }
}
```

OpenMP by Example (reduction)

```
#define SIZE 16000000

int sum = 0;

for (i = 0; i < SIZE; i++) {
    sum += a[i];
}
```

OpenMP by Example (reduction)

```
#define SIZE 16000000

int sum = 0;

#pragma omp parallel for default(shared)
private(i) reduction(+:sum)
for (i = 0; i < SIZE; i++) {
    sum += a[i];
}
```

Assigment

Due Wednesday 21 April