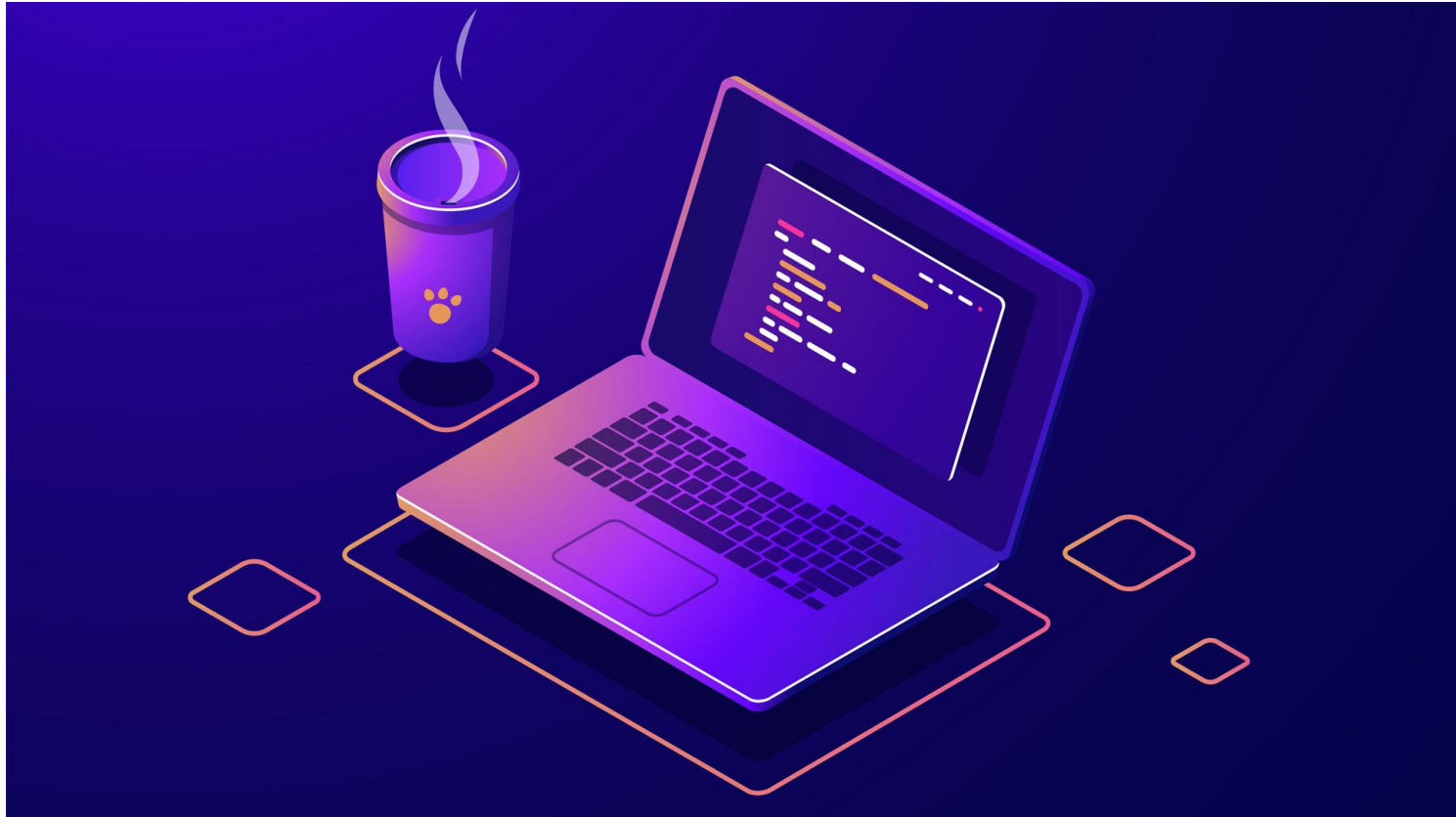


# Introduction to OpenMP





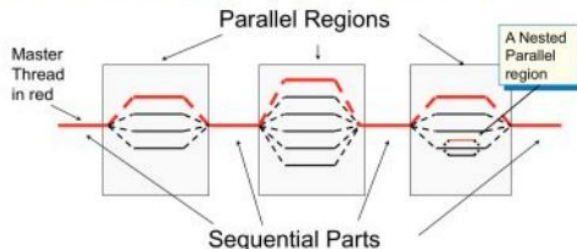
OpenMP is an application programming interface for shared memory multithreaded programming on multiple platforms. It allows adding parallelism to programs written in C, C++ and Fortran.

It consists of a set of compiler directives, library routines, and environment variables that influence runtime behavior.

# Task Creation in OpenMP



- ▶ `#pragma omp parallel`: One **implicit** task is created for each thread in the team (and immediately executed)

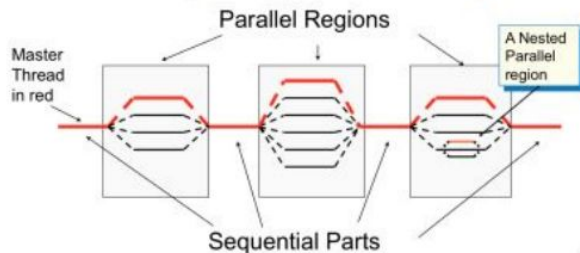


- ▶ `int omp_get_num_threads`: returns the number of threads in the current team. 1 if outside a parallel region
- ▶ `int omp_get_thread_num`: returns the identifier of the thread in the current team that is executing a task, a value between 0 and `omp_get_num_threads() - 1`

# Task Creation in OpenMP



- ▶ `#pragma omp parallel`: One **implicit** task is created for each thread in the team (and immediately executed)



- ▶ `#pragma omp task`: One **explicit** task is created, packaging code and data for (possible) deferred execution
- ▶ `#pragma omp taskloop`: **Explicit** tasks created for chunks of loop iterations
  - ▶ In both cases, tasks executed by threads in the `parallel` region

# Instructor Social Media

**Youtube: Lucas Science**



**Instagram: lucaasbazilio**



**Twitter: lucasebazilio**

