# What is a job system?

A job system manages [multithreaded code](https://docs.unity3d.com/Manual/JobSystemMultithreading.html) by creating [jobs](https://en.wikipedia.org/wiki/Job_(computing)) instead of threads.

A job system manages a group of [worker threads](https://docs.microsoft.com/en-us/cpp/parallel/multithreading-creating-worker-threads) across multiple cores. It usually has one worker thread per [logical CPU core](https://www.howtogeek.com/194756/cpu-basics-multiple-cpus-cores-and-hyper-threading-explained/), to avoid context switching (although it may reserve some cores for the operating system or other dedicated applications).

A job system puts jobs into a [job queue](https://en.wikipedia.org/wiki/Job_queue) to execute. Worker threads in a job system take items from the job queue and execute them. A job system manages [dependencies](http://tutorials.jenkov.com/ood/understanding-dependencies.html) and ensures that jobs execute in the appropriate order.

## What is a job?

A job is a small unit of work that does one specific task. A job receives parameters and operates on data, similar to how a method call behaves. Jobs can be self-contained, or they can depend on other jobs to complete before they can run.

## What are job dependencies?

In complex systems, like those required for game development, it is unlikely that each job is self-contained. One job is usually preparing the data for the next job. Jobs are aware of and support dependencies to make this work. If jobA has a dependency on jobB, the job system ensures that jobA does not start executing until jobB is complete.