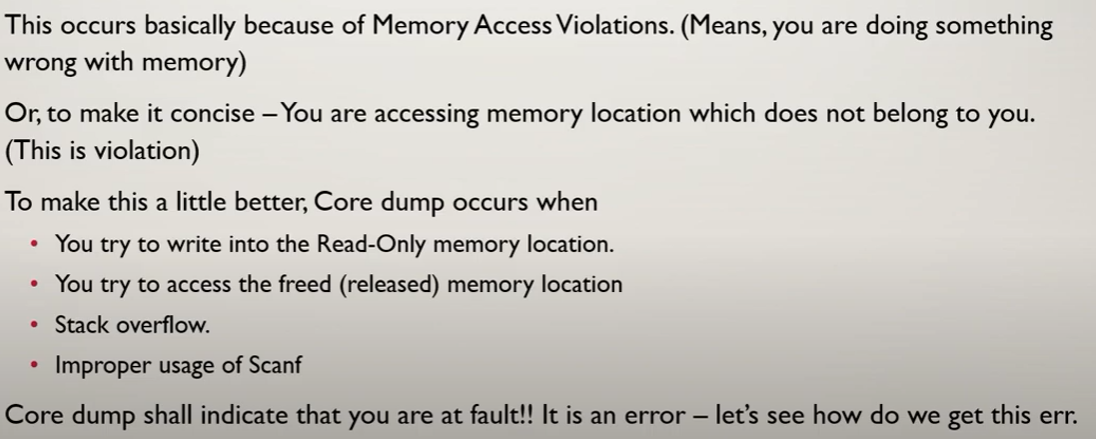
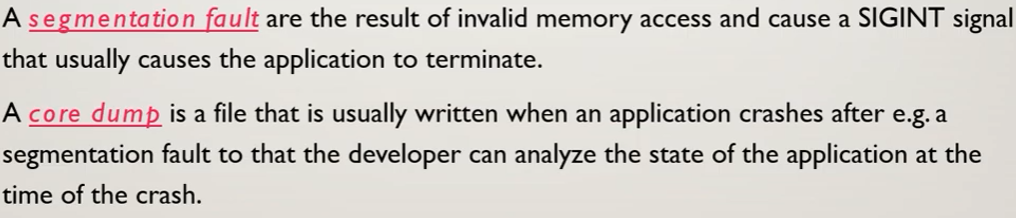
In [computing](https://en.wikipedia.org/wiki/Computing), a **core dump**,[[a]](https://en.wikipedia.org/wiki/Core_dump#cite_note-1) **memory dump**, **crash dump**, **system dump**, or **ABEND dump**[[1]](https://en.wikipedia.org/wiki/Core_dump#cite_note-2) consists of the recorded state of the working [memory](https://en.wikipedia.org/wiki/Computer_storage) of a [computer program](https://en.wikipedia.org/wiki/Computer_program) at a specific time, generally when the program has [crashed](https://en.wikipedia.org/wiki/Crash_(computing)) or otherwise terminated abnormally.[[2]](https://en.wikipedia.org/wiki/Core_dump#cite_note-3) In practice, other key pieces of [program state](https://en.wikipedia.org/wiki/Context_switch) are usually dumped at the same time, including the [processor registers](https://en.wikipedia.org/wiki/Processor_register), which may include the [program counter](https://en.wikipedia.org/wiki/Program_counter) and [stack pointer](https://en.wikipedia.org/wiki/Stack_pointer), memory management information, and other processor and operating system flags and information. A **snapshot dump** (or **snap dump**) is a memory dump requested by the [computer operator](https://en.wikipedia.org/wiki/Computer_operator) or by the running program, after which the program is able to continue. Core dumps are often used to assist in diagnosing and [debugging](https://en.wikipedia.org/wiki/Debugging) errors in computer programs.





The core dump is written in the current directory of the process at the time of the crash.

Of course core dumps need to be enabled, by default those are usually disabled. Check the output of **ulimit -c**, if that's 0 then no core file will be written. Run **ulimit -c unlimited** to enable core dumps; this is a per-process setting which is inherited by processes started by that process.

If a core dump should have been generated but you don't know where, then you could start the process again (if it will without crashing immediately), then check its working directory by doing**ls -l /proc/$pid/cwd** where $pid is the process ID of the process. That link will point to the current working directory of that process. Chances are the core dump will be there. Otherwise you need to run find on the entire system.

Systems using **systemd** are usually configured to dump cores to

**/var/lib/systemd/coredump/**

You may use the **coredumpctl** command to list core dumps. See also [no-more-coredumps-after-migrating-to-systemd](https://unix.stackexchange.com/questions/65110/no-more-coredumps-after-migrating-to-systemd)

**gcore** – команда для снятия дампов процесса

**kdump** – команда для снятия дампов ядра