

RES vs MemStats.Sys memory reporting on Linux

go tool pprof heap

linux top RES

[stackoverflow](#)

Some things to clear. Go is a garbage collected language, which means that memory allocated and used by variables is automatically freed by the garbage collector when those variables become unreachable (if you have another pointer to the variable, that still counts as "reachable").

Freed memory does not mean it is returned to the OS. Freed memory means the memory can be reclaimed, reused for another variable if there is a need. So from the operating system you won't see memory decreasing right away just because some variable became unreachable and the garbage collector detected this and freed memory used by it.

The Go runtime will however return memory to the OS if it is not used for some time (which is usually around 5 minutes). If the memory usage increases during this period (and optionally shrinks again), the memory will most likely not be returned to the OS.

If you wait some time and not allocate memory again, freed memory will be returned to the OS eventually (obviously not all, but unused "big chunks" will be). If you can't wait for this to happen, you may call `debug.FreeOSMemory()` to force this behavior:

FreeOSMemory forces a garbage collection followed by an attempt to return as much memory to the operating system as possible. (Even if this is not called, the runtime gradually returns memory to the operating system in a background task.)