Segments of a Process

- Process' memory is divided into logical *segments*
  - Some are read-only, others read-write
  - Some are known at compile time, others grow dynamically as program runs
- Who assigns memory to segments?
  - *Compiler* and *assembler* generate an *object file* (containing code and data segments) from each *source file*
  - *Linker* combines all the object files for a program into a single executable object file, which is complete and self-sufficient
  - *Loader* (part of OS) loads an executable object file into memory at location(s) determined by the operating system
  - *Program* (as it runs) uses new and malloc to dynamically allocate memory, gets space on stack during function calls