Variables in Assembly Language

Variable Lifetime

	Permanent (Static)	Limited (Dynamic)
Description	variable is always in memory	variable is in memory only when in use
Advantages	maintains value between procedure calls possibly no need to re-initialize (faster)	minimizes memory use may be a register (fast and simple access) usually required for recursive and re-entrant (interrupt) procedures
Disadvantages	always using memory	may be slower (due to allocation and re-initialization) value is lost between procedure calls
When to Use	when needed	whenever possible

Local Variables

Lifetime

Local variables can have either permanent (static) lifetime or limited (dynamic) lifetime. Typically local variables have limited (dynamic) lifetime, but the choice will depend on how the variable is used in the program.

Where to Store

	Register	Stack	Fixed Memory Location
Description	local variable is kept in a register	local variable is stored on stack when enter procedure typically use Y + n to access the variable	local variable is stored at a fixed memory location typically use direct addressing to access the variable
Advantages	fast easy	lots of space available don't need to worry about saving/preserving variables when calling other procedures	easier and possibly faster access than stack lots of space available only way to implement permanent lifetime (static) variables
Disadvantages	limited number of registers only works well for limited lifetime (dynamic) variables may need to save register when calling other procedures	slower access than register variables more complicated access than register variables only works for limited lifetime (dynamic) variables	slower access than register variables static (continuous) use of memory may have conflicts if try to reuse the memory not re-entrant
When to Use	whenever possible	when have many local variables	permanent lifetime (static) variables

Summary

Storage Type	Lifetime	
	Permanent	Limited
Register	Not Recommended	Best Choice
Stack	Not Feasible	if there is a large number of variables
Memory	OK	Not Recommended

Shared Variables

Lifetime

All shared variables have permanent (static) lifetime. Since multiple procedures use the variables they must exist and use memory the entire time the program is executing.

Where to Store

Shared variables must be stored at fixed memory locations as this is the only reasonable way to create a variable with permanent lifetime. Note that a shared variable could be stored in a register, but this would require that the register only be used to store that variable for the entire program. This reduces the number of registers available for other purposes throughout the entire program and thus is not a viable solution.

Assembler Syntax

Definition:

```
    name:   .BYTE      2          ;defines the variable called name to be a word
```

Global Variables

Lifetime

All global variables have permanent (static) lifetime. The variables exist and use memory the entire time the program is executing.

Where to Store

Global variables must be stored at fixed memory locations as this is the only reasonable way to create a variable with permanent lifetime. Note that a global variable could be stored in a register, but this would require that the register only be used to store that variable for the entire program. This reduces the number of registers available for other purposes throughout the entire program and thus is not a viable solution.

Assembler Syntax

A global variable must be defined (**.BYTE** statement) in exactly one source file. Since there is no linker global variables can be accessed from any source file. There is nothing special to do.

Definition:

```
    name:   .BYTE      1          ;defines the variable called name to be a byte
```