A64 Assembler Notes:

* if relocatable output is selected (+r option)then .org statements in the assembler text are ignored. .text is assumed to begin at address zero. This will be adjusted by the linker.
* .text, .rodata, .data, .bss, and .tls sections are output contiguously in the order given beginning at file offset 512 for .elf files or offset zero for a binary file. This is to simplify the loading of files.
* for binary output to work, there can’t be any unresolved externals. All required subroutines must be included in .include files.
* The assembler is currently limited to 10MB per segment.

The register to be updated is to the right. Destination followed by first and second operands. For example ‘add r1,r2,r3’ places the sum of r2 plus r3 into register r1.

Exception is memory store operations. For memory operations the memory operand is always second.

Directives:

.align xxx

* This directive causes the address to be aligned at the given boundary. For example ‘align 16’ would cause the address to advance until the lower digit is a zero (in heximdecimal).

.bss

* sets the segment to the .bss segment. Subsequent output is directed to the bss segment.

.code [start address] to [end address]

.code [number of bits] bits

.code

* Sets the segment to the .code segment. Subsequent output is directed to the code segment. This directive can also establish the number of bits required to represent code addresses.
* Example: code 26 bits
* Example: code 0x8000 to 0xffff

.data

* sets the segment to the .data segment. Subsequent output is directed to the data segment.

.include “<filename>”

* This directive includes another file within the current one.