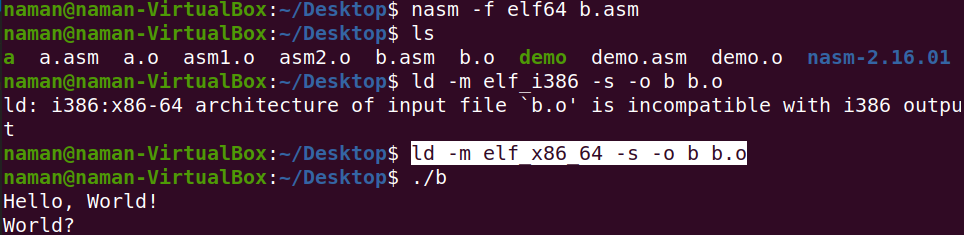
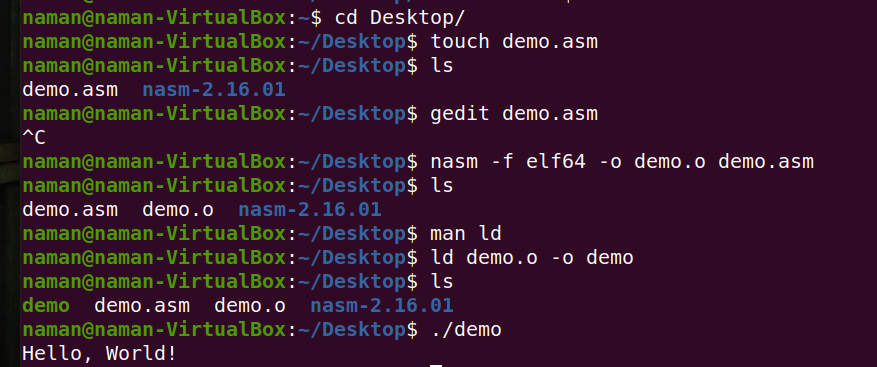
**NASM**

****

**Basic Program :**

****

**First create a asm file**

**Then now write code in it**

**Then compile it using nasm :**

**Nasm -f elf64 -o \_\_\_\_\_\_ \_\_\_\_\_\_\_**

**Now link using linker to a executable file now run executable file**

**GENERAL OVERVIEW**

**What is db in**

**text db “Hello, World!” , 10 (each character is a single byte)**

**text : this is a name assigned to memory that this data is located in , In future the text will be replace with memory address while compiling**

**Db : define bytes**

**It means we are going to define some raw bytes of data to insert in our code**

**10 : new line character**

NASM provides various **define directives** for reserving storage space for variables. The define assembler directive is used for allocation of storage space. It can be used to reserve as well as initialize one or more bytes.

|  |  |  |
| --- | --- | --- |
| **Directive** | **Purpose** | **Storage Space** |
| DB | Define Byte | allocates 1 byte |
| DW | Define Word | allocates 2 bytes |
| DD | Define Doubleword | allocates 4 bytes |
| DQ | Define Quadword | allocates 8 bytes |
| DT | Define Ten Bytes | allocates 10 bytes |

* Each byte of character is stored as its ASCII value in hexadecimal.
* Each decimal value is automatically converted to its 16-bit binary equivalent and stored as a hexadecimal number.
* Processor uses the little-endian byte ordering.
* Negative numbers are converted to its 2's complement representation.
* Short and long floating-point numbers are represented using 32 or 64 bits, respectively.

**CONSTANT\_NAME EQU expression**

The **%assign** directive can be used to define numeric constants like the EQU directive. This directive allows redefinition.

The **%define** directive allows defining both numeric and string constants. This directive is similar to the #define in C.

INC destination

DEC destination

ADD/SUB destination, source