The `section .data` and `section .text` directives define the data and text sections of the program, respectively. The `global _start` directive declares the `_start` label as a global symbol, which means that it can be accessed from other modules. The `_start` label is the entry point of the program, which means that it is the first instruction that is executed when the program starts.

The `mov` instruction is used to move data from one register to another. In this case, the `mov` instructions are used to move the values 1, 1, and 0 into the registers `rax`, `rdi`, and `rsi`, respectively. The `syscall` instruction invokes a system call. In this case, the `syscall` instruction invokes the `write` system call, which writes the string at the address stored in the `rsi` register to the file descriptor stored in the `rdi` register.

The `exit` system call is used to terminate the program. The `exit` system call takes an exit code as its argument. In this case, the exit code is 0, which indicates that the program terminated successfully.

To compile and run this program on Windows, you can use the following commands:

```
nasm -f win64 helloworld.asm
ld -o helloworld helloworld.o
./helloworld
```

The `nasm` command compiles the assembly language file `helloworld.asm` into an object file called `helloworld.o`. The `ld` command links the object file `helloworld.o` into an executable file called `helloworld`. The `./helloworld` command runs the executable file `helloworld`.

I hope this helps! Let me know if you have any other questions.