Inline Assembly Basic

To create the file type the command vi task9.c

Now write this code (to write the code enter I in the keyboard)

static inline uint32\_t rdcycle(void) {

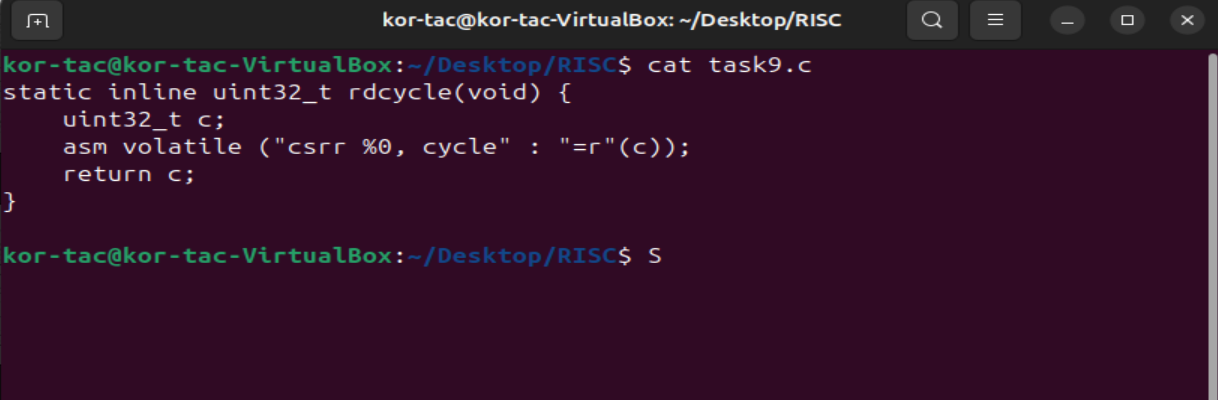
uint32\_t c;

asm volatile ("csrr %0, cycle" : "=r"(c));

return c;

}

Then to view the file use the cat task9.c command



**static inline uint32\_t rdcycle(void)**

* **static**: This limits the function's visibility to the current translation unit. This avoids multiple definitions when this function is included in multiple files.
* **inline**: Suggests to the compiler to replace the function call with the actual function body to reduce function call overhead.
* **uint32\_t**: The return type is a 32-bit unsigned integer.
* **rdcycle**: Function name.
* **void**: Takes no arguments.

**uint32\_t c;**

* Declare a 32-bit unsigned integer c to store the value read from the **cycle counter**.

**asm volatile ("csrr %0, cycle" : "=r"(c));**

This line is the core of the function. Let’s dissect each part:

**🔹 asm / \_\_asm\_\_**

* This is a GCC extension that allows writing **inline assembly**.

**🔹 volatile**

* Tells the compiler **not to optimize away** this assembly instruction.
* This is important because the **cycle counter changes constantly**, and we want to ensure this read is not skipped or moved around.

**🔹 "csrr %0, cycle"**

* This is the actual **RISC-V assembly instruction**.
* csrr (Control and Status Register Read) copies the content of a CSR (here, the cycle counter cycle, which is at CSR address 0xC00) into a general-purpose register.
* %0 is a **placeholder** for the first output operand. It gets substituted by a register (chosen by the compiler) where the cycle count will be stored.

**🔹 : "=r"(c)**

This part is called the **output operand constraint**.

Let’s break it down:

* **=**: Means the operand is **write-only**. The assembly will output to this operand.
* **r**: Means any **general-purpose register** can be used.
* **(c)**: Tells the compiler to map the value from the register (after csrr) into the C variable c.

| **Part** | **Meaning** |
| --- | --- |
| "csrr %0, cycle" | Inline RISC-V assembly to read CSR 0xC00 (cycle counter) |
| volatile | Prevents compiler from optimizing out or reordering the asm code |
| =r | Output-only (=), general-purpose register (r) |
| (c) | Output will be stored in the variable c |