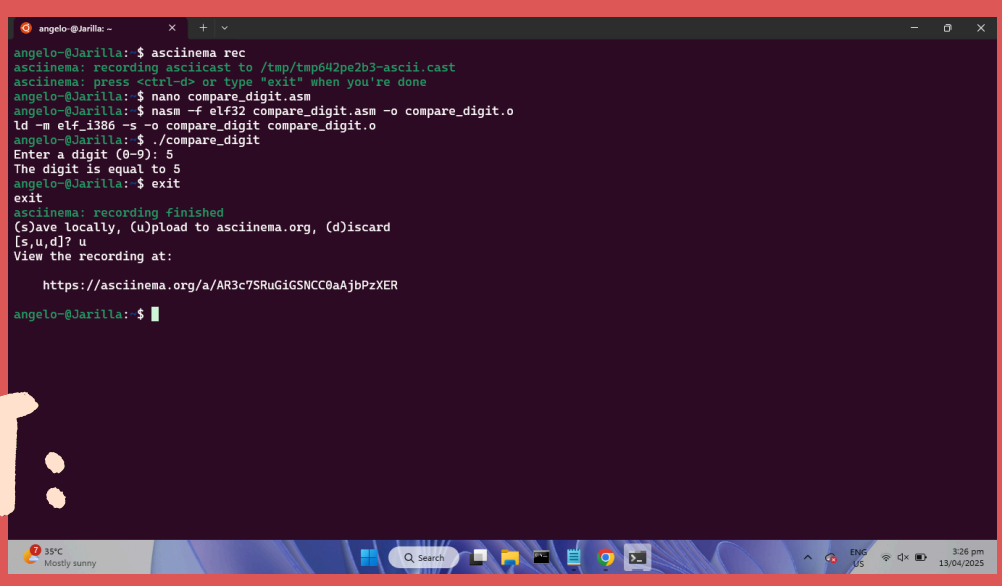
TASK PERFORMANCE

JARILLA, ANGELO CHRISTIENE M.

OUTPUT:



COMPARE_DIGIT.ASM

```
section .data
    prompt db "Enter a digit (0-9): ", 0
    below_msg db "The digit is below 5", 10, 0
    equal_msg db "The digit is equal to 5", 10, 0
    above_msg db "The digit is above 5", 10, 0

section .bss
    input resb 2

section .text
    global _start

_start:
    mov eax, 4
    mov ebx, 1
    mov ecx, prompt
    mov edx, 22
    int 0x80

    mov eax, 3
    mov ebx, 0
    mov ecx, input
    mov edx, 2
    int 0x80

    movzx eax, byte [input]
    sub eax, '0'

    call compare_digit

    mov eax, 1
    xor ebx, ebx
    int 0x80

compare_digit:
    cmp eax, 5
    jl below
    je equal
    jg above

below:
    mov eax, 4
    mov ebx, 1
    mov ecx, below_msg
    mov edx, 23
    int 0x80
    ret

equal:
    mov eax, 4
    mov ebx, 1
    mov ecx, equal_msg
    mov edx, 24
    int 0x80
    ret

above:
    mov eax, 4
    mov ebx, 1
    mov ecx, above_msg
    mov edx, 23
    int 0x80
    ret
```

EXPLANATION:

The program begins by prompting the user to enter a digit from 0 to 9, which is read from standard input. It then converts the ASCII input to a numerical value by subtracting the ASCII value of '0'. The core logic of the comparison is encapsulated in a procedure named compare_digit, which uses cmp and conditional jumps (jl, je, jg) to determine whether the input is below, equal to, or above 5. Based on the result, it prints the corresponding message. The use of syscall interrupts (int 0x80) ensures communication with the Linux system for both input and output. The structure demonstrates procedural calling and branching effectively.