Lab06 Report

崔士强 PB22151743

December 31, 2023

1 Purpose

The purpose of the program is to compute the factorial of a non-negative number N,

Anticipated outcomes: The program starts with repeatedly printing the student ID, waiting for the input. For diffent kinds of input, the output varies. First the program tells whether the input is a decimal digit. If yes, the output is as follows:

1. $0 \leq N \leq 7:$ $N!$ is displayed.
2. $8 \leq N \leq 9:$ print: "$N!$ is too large for LC-3."

2 Principles

2.1 Calling ISR

In the OS code, bit 14 of the KBSR is set to 1. When input is detected, bit 15 of the KBSR is set to 1, enabling the interrupt signal. Then the interrupt service routine is called.

2.2 Getting literal value of input

When we use `GETC` to get the input character, the ASCII value is stored in `R0`. We also need its original value.

During checking the range of input, the instructions below are performed:

```
LD      R1, ASCII_0
ADD     R1, R0, R1
BRn     NON_DEC       ; Smaller than '0'
```

Here `ASCII_0` is `#-48`. After the `ADD` instruction, the value in R1 is the literal value of the input. So we store it in another register for possible future use.

2.3 Giving the results

In this program, a trivial approach is used to give the final result: list all possibilities:

```
          ADD    R3, R3, #-1
          BRn    RES0
          ADD    R3, R3, #-1
          BRn    RES1
          ; RES2 through RES7 are similar and not displayed here
RES0      LEA    R0, MSG_RES0
          PUTS
          BRnzp  RETURN
RES1      LEA    R0, MSG_RES1
          PUTS
          BRnzp  RETURN
```

3 Procedure

3.1 Bugs encountered

When testing the program, I saw ”Access violation” in the console. After debugging, I found out that the bug was caused by inappropriate instruction to print strings. To print strings using PUTS, the starting address of the string must be loaded into `R0`. Example:

```
  LEA    R0, MSG_EXCD
  PUTS
  MSG_EXCD  .STRINGZ  "! is too large for LC-3."
```

But I mistakenly used `LD`.

Solution: use `LEA` instead.

3.2 Chanllenges

A major challenge in the task is to print the result.

Reason for its difficulty is that the result may be more than 1 digit. For a number like this, we can not directly convert it to ASCII value. Possible procedures to print the result are as follows:

1. Divide the result by 10
2. Obtain the remainder and substract this remainder from the result
3. Save the remainder
4. Repeat step 1 if the result is positive
5. Print the remainders in reverse order

Due to limited time, this approach, though more like “computing”, is not adopted.

4 Results

Screen recording: Lab06 Program Test