ICS_Lab3_Report

Xiaoma

2022 年 11 月 29 日

实验目的

使用 LC-3 汇编命令实现求字符串的最大重复子串。
重复子串是一种由相同字符组成的子串。例如，“aabbbc”的重复子串为“aa”、“bbb”和“c”。

- 字符串的长度 N 存储在内存位置 x3100
- 字符串的每个字符存储在以内存位置 x3101 开始的连续内存位置中
- 将最长重复子串的长度存储在内存位置 x3050

实验原理

对于求最大重复子串问题，通常通过一次遍历来得到正确结果，即程序存储此时最大重复子串的长度以及当前重复子串的长度，若此时当前重复子串结束，则将其长度与最大长度比较，更新最大值。
根据该思想可以得到伪代码：

Algorithm 1: maxRepeating

Input: The string : str; Size of str : N;

Output: The max-size of repetitive substring : max_len;

```
right = str[0];
left = right;
max_len = 1;
temp = 1;
N− = 1;
i = 1
while N > 0 do
    right = str[i];
    i+ = 1;
    if left == right then
        temp+ = 1;
    else
        if max_len < temp then
            max_len = temp;
        temp = 1;
    left = right;
    N− = 1;
return max_len;
```

实验步骤

LC-3 指令集的限制

在之前的实验中对于 BR, ST 等指令通常直接使用十进制或十六进制偏移量，这样会使设计程序时计算地址变得繁琐，故在本次实验中将使用 LABEL 来降低计算量。

边界问题

- 当 $N = 1$ 时：
  此时最长重复子串始终为 1，故程序开始时应将此时最长重复字串长度初始化为 1。
- 当字符串中最后一个字符与其前一个相同时：
  此时应先比较最后一个重复子串的长度更新最大值，而不应该直接将最大值存入 x3050。

代码讲解

初始化变量

从内存中读取字符串的长度 N，初始化字符串指针，即

$$R0 \leftarrow N, R1 \leftarrow \&str$$

```
1    INIT  LDI  R0, NUM
2          LD  R1, DATA
```

处理 $N = 1$ 的边界问题

将此时最大的重复子串长度初始化为 1，读取字符串中第一个字符，指针向后移动一位，即

$$R2 \leftarrow mem[R1], R3 \leftarrow R2, R5 \leftarrow 1$$

```
3     LDR R2, R1, #0
4     ADD R3, R2, #0
5     ADD R1, R1, #1
6     ADD R5, R5, #1
7     ADD R4, R4, #1
8     ADD R0, R0, #-1
```

遍历字符串

遍历字符串，比较遍历到的字符与其前一个字符是否相同，若相同则该子串长度加一，若不同则跳转至 UPDATE，若循环结束则跳转至 UPDATE 进行最后一次更新。

```
9     WHILE BRz UPDATE
10        LDR R2, R1, #0
11        ADD R1, R1, #1
12        NOT R6, R3
13        ADD R6, R6, #1
14        ADD R6, R2, R6
15        BRnp UPDATE
16        ADD R4, R4, #1
17    BACK ADD R3, R2, #0
18        ADD R0, R0, #-1
19        BRnzp WHILE
```

更新最大值

比较子串长度与当前最大重复子串长度，取较大值更新最大长度，并将子串长度重置为 1，判断此时是否结束遍历，若结束则跳转至 OUTPUT，若未结束则返回 WHILE。

```
20    UPDATE NOT R6, R4
21        ADD R6, R6, #1
22        ADD R6, R5, R6
23        BRzp #1
24        ADD R5, R4, #0
```

```
25          AND R4, R4, #0
26          ADD R4, R4, #1
27          ADD R0, R0, #0
28          BRz OUTPUT
29          BRnzp BACK
```

存储结果

将最终结果存储到 x3050

```
30      OUTPUT STI R5, RESULT
```

实验结果

依次对实验文档给出的例子进行测试，结果如下：

汇编评测

3 / 3 个通过测试用例

- 平均指令数: 77
- 通过 6:aabbbc:3, 指令数: 83, 输出: 3
- 通过 5:ZZZZz:4, 指令数: 65, 输出: 4
- 通过 6:aabaaa:3, 指令数: 83, 输出: 3

自行编写了部分测试例子，结果如下：

汇编评测

3 / 3 个通过测试用例

- 平均指令数: 250.33333333333334
- 通过 10:aaabbbcdfr:3, 指令数: 158, 输出: 3
- 通过 15:abcddddfffrrrrr:5, 指令数: 214, 输出: 5
- 通过 20:asdfghjklqwertyuiopz:1, 指令数: 379, 输出: 1