

# 山东大学 计算机科学与技术 学院

## 汇编语言 课程实验报告

学号：202120130276	姓名：王云强	班级：21.2 班
实验题目：实验 6：例 3.8，例 4.1		
实验学时：2	实验日期：2023.11.24	
实验目的：熟悉实验环境、调试学习课本示例		
<p>实验环境：</p> <ol style="list-style-type: none"><li>1. 以键盘读取文件操作为例，掌握进行 BIOS 与 DOS 调用的思路。</li><li>2. 掌握 DOS 下文件的读取、写入与显示，以及使用 BIOS 读取键盘的方法。掌握输入缓冲区的基本设计与使用方法；熟悉常规命令行程序的人机交互逻辑编写。</li></ol>		
<p>源程序清单：</p> <ol style="list-style-type: none"><li>1. WSPP.ASM（示例 3.8 源程序）</li><li>2. EX_41.ASM（示例 4.1 源程序）</li></ol>		
<p>编译及运行结果：</p> <p>示例 3.8 编译结果：</p> <pre>C:\&gt;masm wspp Microsoft (R) Macro Assembler Version 5.00 Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.  Object filename [wspp.OBJ]: Source listing [NUL.LST]: Cross-reference [NUL.CRF]:  51562 + 448598 Bytes symbol space free  0 Warning Errors 0 Severe Errors  C:\&gt;link wspp  Microsoft (R) Overlay Linker Version 3.60 Copyright (C) Microsoft Corp 1983-1987. All rights reserved.  Run File [WSPP.EXE]: List File [NUL.MAP]: Libraries [LIB1]: LINK : warning L4021: no stack segment</pre>		

示例 3.8 运行结果：

未插入前：

```
abcdefg  
hi jklmn  
opq
```

插入后：

```
abcdefg  
hi jklmn  
opabcdefgg
```

示例 4.1 编译结果：

```
C:\>masm ex_41  
Microsoft (R) Macro Assembler Version 5.00  
Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.  
  
Object filename [ex_41.OBJ]:  
Source listing [NUL.LST]:  
Cross-reference [NUL.CRF]:  
  
51602 + 448558 Bytes symbol space free  
  
0 Warning Errors  
0 Severe Errors  
  
C:\>link ex_41  
  
Microsoft (R) Overlay Linker Version 3.60  
Copyright (C) Microsoft Corp 1983-1987. All rights reserved.  
  
Run File [EX_41.EXE]:  
List File [NUL.MAP]:  
Libraries [LIB]:  
LINK : warning L4021: no stack segment
```

示例 4.1 运行结果：

File.txt 内容为单行时：

File.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

sducs

```
C:\>EX_41
```

```
    Please input filename: File.txt
sducs
```

File.txt 内容为多行时:

File.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

123456

sducs

456789

sducs

```
C:\>EX_41
```

```
    Please input filename: File.txt
123456
sducs
456789
sducs
```

问题及收获：

### 1. 对源程序 3.8 的代码认识：

①在示例 3.8 源程序中有一个 curs 宏，传入两个参数 row 和 col，传参方式是全局变量作为参数传入，作用是调用 10H 号中断的 2 号功能，将光标位置放在第 0 页的 row 行 col 列，该宏的功能是将光标置位到指定位置。

②在 3.8 中有一个子程序 DISPBF，无参数，其主要作用为将内存中 KBD\_BUF 数组中的所有内容全部打印出来，主要通过 10H 号中断的 0EH 号功能，将 AL 打印到屏幕上（BL 为 0，控制背景色）。

③整体框架：首先先初始化所有的指针，清空屏幕，之后通过 16H 号中断的 0 号功能读键盘输入到 AL 中，之后判断输入的内容，分为三种情况，方向键、字符、回车。

首先是最简单的字符，如果首尾指针重合，则直接插入，否则则将时头指针到尾指针的所有内容向后移动一个单位，再插入字符。

之后是回车，需要将指针后面的内容全部向下移动一行，调整行、列指针位置。

最后是方向键。

对于左方向键，首先先判断是不是第 0 列，如果是，再判断是不是第 0 行，如果是第 0 行第 0 列，则不动。若是第 0 列，不是第 0 行，则指向 上一行的该列位置。若不是第 0 行，则直接列数-1，最后都要重置光标位置。

对于右方向键，与左方向键的处理方法大致一样，只不过在这里末尾成

了左方向键中的第 0 行第 0 列（开头）的位置，下一个字符是不是回车成了左方向键中的是不是第 0 列的判断。

## 2. 对于源程序 4.1 有 BUG，哪里有 BUG，如何修改使之正确运行？

BUG 原因是因为文件名也放在了文件内容的缓冲区（BUF）内。导致输入的内容与文件名发生混乱，无法正确判断是否读取到文件的末尾，同时源代码中判断文件结尾是以 1AH 来判断的，但实际文件结尾应该是以 00 为结尾的，所以这个地方也需要改。

**即两个 BUG：**

- ①如果文件内容长度短于文件名字，则读取到的文件内容实际为文件内容和文件名字未被覆盖的部分；
- ②不能正确判断文件结尾，当文件读入为 00 时，为真正的文件结尾。

**解决方法：**需要重新定向文件内容指针使其从文件真正内容开始读，不将文件名字也视作内容，之后只要读到 0，说明文件内容结束，正常退出即可。落实到代码上，也就是首先修改文件内容读取部分，之前是最多读入 200 字节内容，现在由于文件名（假设长度为  $x$ ）也在其中，所以现在需要 先修改指针，使得最多读入  $200-x$  的字节的内容。之后是输出部分的代码，之前是从文件的开头开始输出，现在需要每一次输出前，先移动指针，使指针从第  $x$  个单元开始读取（即跳过文件名），这样就开始只读文件内容了。而在判断文件是否读完时，利用先前示例程

序提供的基础，只需要判断是不是读到了 0 即可，如果是 0，则说明文件内容结束，直接退出即可。

### 3. 对于源程序 4.1 的认识：

首先对于该程序有许多子函数，需要一一简单说明。

#### ①CHANGE\_PSIZE 子函数：

无参数，无返回值。主要功能是在要求用户输入一个值，将该值转换成真实数字，用来定义更改之后，一页显示多少行，行数不得超过 24，也不得小于 0。

#### ②OPENF 子函数：

传入一个参数作为文件名，以全局变量形式传入，返回一个 AX 的值，作为文件是否打开的判断标识，如果是 0 则打开失败，否则打开成功。主要功能是打开文件名对应的文件，并给予一个是否成功打开的反馈。

#### ③GETLINE 子函数：

不传参，也不返回值，主要功能就是与用户交互，让用户输入文件名，并将 0 放在文件名后面。

#### ④READ\_BLOCK 子函数：

传入全局变量作为参数，不返回值。主要功能是从 BX（HANDLE 这一全局变量）（存着文件名）中打开文件，之后读取 CX（默认是 200 字节，

但个人认为应该修改为 200-文件名所占字节数)对应的字节到 BUF 中，也就是实验教程中所提到的读取至多 200 个字节到 BUF（文件内容缓存）中。

#### ⑤SHOW\_BLOCK 子函数：

个人感觉这个子函数应该不算是传参，但有返回值。主要功能是读取一行的内容，先校对指针，从现在文件应该读取的地方开始读取字符，逐个进行显示。如果读到了文件末尾，则使得 BX=0，进行返回反馈，方便主函数判断当前是否读到了文件的末尾。

#### ⑥主函数流程：

首先先获取文件名，之后再打开对应的文件，进行读取，至多 12 行内容（如果不够 12 行，读完、输出即可结束，够 12 行，则在输出完 12 行内容后，再输出一行星号），之后判断用户的下一次输入的字符，如果是空格，则继续读取 12 行，如果不是空格、而是 p，则让用户输入修改之后一页最多多少行，后续内容根据这个值进行输出。举个例子，一共有 17 行内容，先读取前 12 行输出，之后如果再输入空格，则输出完后 5 行结束；如果读取前 12 行并输出后，输入 p，则修改一页的行数（假设修改为 2），之后输出第 13-14 行内容，之后再按 p 无效，只有按空格，才会输出 2 行内容，直到输出完所有文件的内容。