S&T2024 Advanced C Programming Lab 2

Compression for Numerical Data and Binary Files

18 May 2025 Wednesday 6:50pm

Unzip **AC_Lab2.zip**.

The use of binary files allows for data compression across byte boundaries. This is useful in saving space, and is even more efficient in saving time. On a machine with 32-bit words, each instruction cycle processes 32 bits in parallel. If more data are packed in a word, the speed of data processing will be increased proportionately. In this practical we are going to write a program to compress a text file that contains only digits and reduce its storage requirement by half.

We first analyze the approach used in this assignment. Each digit in the text file is stored as a character and therefore it requires one byte (1 byte = 8 bits). Actually, half a byte (4 bits) is sufficient to keep the contents of the digit as you can observe from the following table:

Digit	Binary Representation
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

To compress the text file we shall proceed as follows. We shall pack every eight characters into one 32-bit word. In our example using unsigned integer, one word contains 4 bytes (and 1 byte contains 8 bits), giving 32 bits. To simplify this assignment we assume that the number of digits in the text file is a multiple of 8. For each digit (character) we only keep the 4 least significant bits and pack them in a 32-bit **`unsigned integer`**. In this way the 32-bit word can keep the contents of up to eight digits as shown in the following example.

8 digits

Suppose the digits are `31561234`. The binary contents of the compressed data will be `0011 0001 0101 0110 0001 0010 0011 0100`.

32 bits

The text file you are going to compress is named as **digit.inf**.

Contents of digit.inf

```
66677266661234567890123456789077266666772
```

The compressed data will be created by your program and stored in a binary file named as **digit.ouf** by the **fwrite** function. After you have run your program, the following file sizes should be displayed if you type the **DIR** command at the command prompt. You discover that the file size of **digit.inf** can be reduced from 40 bytes to 20 bytes.

```
D:\>dir digit.*
 Volume in drive D is New Volume
 Volume Serial Number is BE71-0209

 Directory of D:\

14/06/2020  01:29 pm                40 digit.inf
14/06/2020  03:29 pm                20 digit.ouf
               2 File(s)             60 bytes
               0 Dir(s)  194,832,150,528 bytes free
```

To verify that the compression is done correctly, your program should subsequently use the **fread** function to read and decompress the contents in **digit.ouf** as shown on screen:

```
C:\Windows\system32\cmd.exe


   Original Text: 66677266661234567890123456789077266666772
Decompressed Text: 66677266661234567890123456789077266666772

Press any key to continue . . .
```

When typing out on the command prompt, the contents of **digit.ouf** should be as follows if your program is correct:

```
D:\>type digit.ouf
frgfV4□f4□ÉxwÉxVrgf&
```

- End -