

แผนการสอนประจำบทที่ 10

การประมวลผลเพิ่มข้อมูล (File Processing)

หัวข้อสำคัญ

1. ลำดับของข้อมูล (data hierarchy)
2. วิธีการติดต่อกับเพิ่มข้อมูลการจองพื้นที่ในหน่วยความจำ
3. ประเภทของไฟล์และการทำงานกับเพิ่มข้อมูล
4. ฟังก์ชันที่ใช้เกี่ยวกับการประมวลผลเพิ่มข้อมูล

วัตถุประสงค์เชิงพฤติกรรม

1. ผู้เรียนสามารถอธิบายการทำงานกับเพิ่มข้อมูลได้
2. ผู้เรียนสามารถเขียนโปรแกรมการทำงานกับเพิ่มข้อมูลได้

วิธีการสอนและกิจกรรมการเรียนการสอน

1. การบรรยาย
2. การทำแบบฝึกหัด

สื่อที่ใช้ประกอบการสอน

1. เอกสารประกอบการสอน
2. เครื่องคอมพิวเตอร์
3. เครื่องฉายภาพนิ่ง

การวัดและประเมินผล

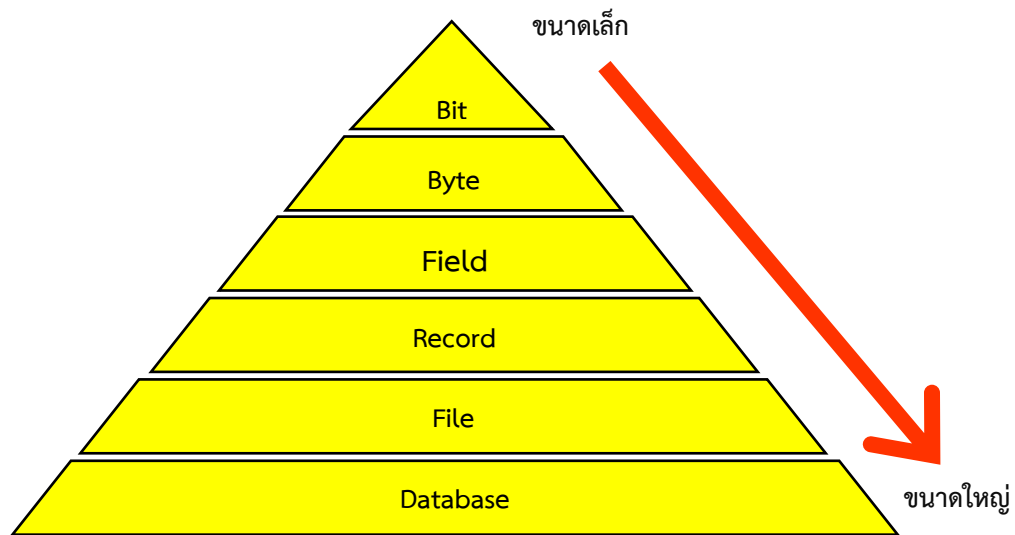
1. สังเกตจากความสนใจของผู้เรียน
2. ประเมินจากการตอบคำถามของผู้เรียนและกิจกรรมในชั้นเรียน
3. การทำแบบฝึกหัดท้ายบท

บทที่ 10

การประมวลผลเพิ่มข้อมูล (File Processing)

10.1. ลำดับของข้อมูล (data hierarchy)

ลำดับของข้อมูลเรียงลำดับตามขนาดของการเก็บข้อมูลแสดงดังภาพ 10.1

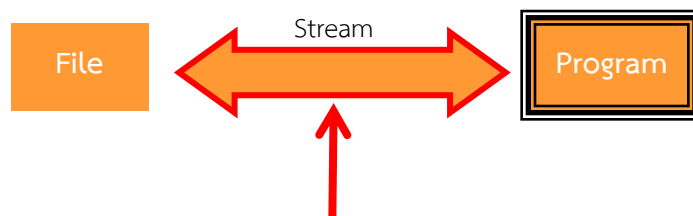


ภาพ 10.1 จำลองการลำดับและขนาดข้อมูล

10.2. วิธีการติดต่อกับเพิ่มข้อมูล

Stream เป็นตัวเชื่อมต่อระหว่างเพิ่มข้อมูลกับโปรแกรมเพื่อรับและส่งข้อมูลระหว่างกันแสดงดังภาพ

10.2



ข้อมูลจะมีการไหลเข้า-ออกผ่าน Stream

ภาพ 10.2 การส่งข้อมูลแบบ Stream

10.3. ประเภทของไฟล์

10.3.1 Text files

เป็นไฟล์ที่เก็บข้อมูลเป็นตัวอักษร ตัวเลข และสัญลักษณ์พิเศษได้ สามารถสร้างไฟล์จากโปรแกรมจากโปรแกรมง่าย เช่น notepad, ediplus ตัวอย่างนามสกุล .txt, .dat

10.3.2 Binary files

เป็นไฟล์ที่เก็บข้อมูลเป็นเลขฐาน 2 คือ 0 และ 1 เท่านั้น ส่วนใหญ่จะมีนามสกุล .bin

10.4. การทำงานกับแฟ้มข้อมูล

การทำงานกับแฟ้มข้อมูล หรือ ไฟล์ นั้น ต้องอาศัยตัวแปรพอยน์เตอร์ (Pointer) เป็นตัวชี้ไปยังแฟ้มข้อมูลทำงาน ต้องกำหนดให้พอยน์เตอร์ดังกล่าวมีชนิดข้อมูลเป็นไฟล์ โดยชื่อตัวแปรเป็นชื่อของตัวแปรพอยน์เตอร์ที่ใช้ชี้ไปยังแฟ้มข้อมูล

| รูปแบบการประกาศตัวแปรไฟล์ | ตัวอย่าง |
|---------------------------|--------------------------------|
| FILE *ชื่อตัวแปร; | FILE *fptr; FILE *file_ptr; |

10.5 การเปิด-ปิดแฟ้มข้อมูล

หลังจากมีการประกาศตัวแปรตัวแปรพอยน์เตอร์ตัวชี้ไปยังไฟล์แล้วนั้น เมื่อต้องการเรียกใช้งานไฟล์ต้องระบุตำแหน่งและชื่อไฟล์ที่ต้องการเรียกงานพร้อมกับระบุรูปแบบการทำงาน (**mode**) ให้กับไฟล์ด้วย รูปแบบการทำงานของแฟ้มข้อมูลแบ่งตามประเภทไฟล์ Text file และ Binary file แสดงดังต่อไปนี้

10.5.1 รูปแบบการทำงานของแฟ้มข้อมูลแบ่งตามประเภทไฟล์ Text file

| Mode | คำอธิบาย |
|-----------------------|--|
| การทำงานกับ Text file | |
| r | เปิดไฟล์ |
| w | เปิดไฟล์และเขียนไฟล์ (ถ้าไม่มีไฟล์อยู่จะสร้างไฟล์ใหม่) |
| a | เปิดไฟล์และเขียนไฟล์ต่อจากไฟล์เดิม (เขียนต่อท้าย) |

| | |
|----|--|
| r+ | เปิดไฟล์โดยถ้าไม่มีแฟ้มเดิมอยู่จะทำงานไม่ได้ |
| w+ | เปิดไฟล์และเขียนไฟล์โดยถ้าไม่มีแฟ้มเดิมอยู่จะทำงานไม่ได้ |
| a+ | เปิดไฟล์และเขียนไฟล์ต่อจากไฟล์เดิม (เขียนต่อท้าย) โดยถ้าไม่มีแฟ้มเดิมอยู่จะทำงานไม่ได้ |

10.5.2 รูปแบบการทำงานของแฟ้มข้อมูลแบ่งตามประเภทไฟล์ Binary file

| Mode | คำอธิบาย |
|-------------------------|--|
| การทำงานกับ Binary file | |
| rb | เปิดไฟล์ |
| wb | เปิดไฟล์และเขียนไฟล์ (ถ้าไม่มีไฟล์อยู่จะสร้างไฟล์ใหม่) |
| ab | เปิดไฟล์และเขียนไฟล์ต่อจากไฟล์เดิม (เขียนต่อท้าย) |
| rb+ | เปิดไฟล์ โดยถ้าไม่มีแฟ้มเดิมอยู่จะทำงานไม่ได้ |
| wb+ | เปิดไฟล์และเขียนไฟล์โดยถ้าไม่มีแฟ้มเดิมอยู่จะทำงานไม่ได้ |
| ab+ | เปิดไฟล์และเขียนไฟล์ต่อจากไฟล์เดิม (เขียนต่อท้าย) โดยถ้าไม่มีแฟ้มเดิมอยู่จะทำงานไม่ได้ |

10.5.3 คำสั่งการเปิดไฟล์

คำสั่งการเปิดไฟล์จะใช้ fopen เพื่อใช้คำสั่งเปิดไฟล์ คำสั่งนี้มี พารามิเตอร์ที่คำสั่ง 2 พารามิเตอร์ ได้แก่

ตำแหน่งและชื่อไฟล์ = ใช้ระบุตำแหน่ง, ชื่อ-นามสกุลของแฟ้มข้อมูลภายในคอมพิวเตอร์ ถ้าไม่ระบุตำแหน่งของแฟ้มข้อมูล ภาษาซีจะค้นหาหรือสร้างไฟล์ที่ตำแหน่งเดียวกับ Source code

รูปแบบการทำงาน (mode) = กำหนดฟังก์ชันการทำงานกับแฟ้มข้อมูล เช่น r คือ อ่านไฟล์, w คือ เขียนไฟล์

| รูปแบบคำสั่งการเปิดไฟล์ |
|---|
| ตัวแปรพอยเตอร์ = <code>fopen("ตำแหน่งและชื่อไฟล์", "รูปแบบการทำงาน")</code> |
| ตัวอย่างที่ 1 |
| 1. <code>FILE *fptr;</code> 2. <code>fptr = fopen("test.txt", "w");</code> |
| ตัวอย่างที่ 2 |
| 1. <code>FILE *fptr;</code> 2. <code>fptr = fopen("c:\\cprogram\\test.txt", "r");</code> |

10.5.4 คำสั่งการปิดไฟล์

หลังจากประมวลผลไฟล์เสร็จเรียบร้อยแล้ว ต้องยกเลิกการเชื่อมต่อกับไฟล์โดยใช้คำสั่ง `fclose`

| รูปแบบคำสั่งการปิดไฟล์ | ตัวอย่าง |
|--------------------------------------|--|
| <code>fclose(ตัวแปรพอยเตอร์);</code> | <code>fclose(fptr);</code> <code>fclose(file_ptr);</code> |

10.6 การเขียนลงแฟ้มข้อมูล

10.6.1 การเขียนลงแฟ้มข้อมูลด้วยคำสั่ง `fprintf()`

การเขียนลงแฟ้มข้อมูลใช้คำสั่ง `fprintf()` โดยรูปแบบการใช้คำสั่งคล้ายกับคำสั่ง `printf()` ปกติ คือ ต้องระบุข้อความ หรือ รูปแบบข้อมูลที่ต้องเขียนไว้ภายในคำสั่ง

| รูปแบบคำสั่งการเขียนลงแฟ้มข้อมูลด้วยคำสั่ง <code>fprintf()</code> |
|---|
| <code>fprintf(ตัวแปรพอยน์เตอร์, "ข้อมูลที่เขียน");</code> หรือ <code>fprintf(ตัวแปรพอยน์เตอร์, "ข้อมูลที่เขียน+การควบคุมการแสดงผล", ตัวแปร);</code> |

ตัวอย่าง

1. `FILE *fptr;`
2. `int number = 2;`
3. `fptr = fopen("test.txt","w");`
4. `fprintf(fptr, "File processing\n");`
5. `fprintf(fptr, "Line number = %d", number);`

ตัวอย่างการเขียนลงแฟ้มข้อมูลด้วยคำสั่ง fprintf()

โปรแกรม

1. `#include <stdio.h>`
2. `main()`
3. `{`
4. `FILE *fptr;`
5. `char text[25];`
6. `int index;`
7. `fptr = fopen("input.txt","w");`
8. `strcpy(text,"*---Line---*");`
9. `fprintf(fptr,"%s\n", text);`
10. `for (index = 1; index <= 10; index++)`
11. `fprintf(fptr,"Line number is %d\n", index);`
12. `fclose(fptr);`
13. `printf("Finished");`
14. `}`

| ผลลัพธ์ |
|---------|
| |

10.6.2 การเขียนลงแฟ้มข้อมูลด้วยคำสั่ง fputc()

เป็นคำสั่งที่ใช้เขียนอักขระลงในแฟ้มข้อมูล

| รูปแบบคำสั่งการเขียนลงแฟ้มข้อมูลด้วยคำสั่ง fputc() |
|--|
| <pre>fputc(ตัวแปร, ตัวแปรพอยน์เตอร์); หรือ fputc('ตัวอักขระ', ตัวแปรพอยน์เตอร์);</pre> |
| ตัวอย่าง |
| <pre>1. FILE *fptr; 2. char ch= 'A'; 3. fptr = fopen("test.txt","w"); 4. fputc(ch, fptr); 5. fputc('Z', fptr);</pre> |

ตัวอย่างการเขียนลงแฟ้มข้อมูลด้วยคำสั่ง fputc()

| โปรแกรม |
|--|
| <pre>1. #include <stdio.h> 2. main() 3. { 4. FILE *fp; 5. char ch ='y'; 6. int i; 7. fp = fopen("input-fputc.txt", "w+"); 8. for(i = 0 ; i <= 5; i++) 9. fputc(ch, fp); 10. for(ch = 65 ; ch <= 90; ch++) 11. fputc(ch, fp); 12. printf("Finished"); 13. fclose(fp); 14. }</pre> |
| ผลลัพธ์ |
| |

10.6.3 การเขียนลงแฟ้มข้อมูลด้วยคำสั่ง fputs()

เป็นคำสั่งที่ใช้เขียนข้อความลงในแฟ้มข้อมูล

| รูปแบบคำสั่งการเขียนลงแฟ้มข้อมูลด้วยคำสั่ง fputs() |
|--|
| <pre>fputs(ตัวแปร, ตัวแปรพอยน์เตอร์);</pre> <p>หรือ</p> <pre>fputs('ข้อความ', ตัวแปรพอยน์เตอร์);</pre> |
| ตัวอย่าง |
| <pre>1. FILE *fptr; 2. char ch[10]= "ABC"; 3. fptr = fopen("test.txt","w"); 4. fputs(ch, fptr); 5. fputs("DEF", fptr);</pre> |

ตัวอย่างการเขียนลงแฟ้มข้อมูลด้วยคำสั่ง fputs()

| โปรแกรม |
|--|
| <pre>1. #include <stdio.h> 2. main() 3. { 4. FILE *fp; 5. char ch[50] = "Hello student"; 6. int i; 7. fp = fopen("input-fputs.txt", "w"); 8. for(i = 0 ; i <= 5; i++) 9. { 10. fputs(ch, fp); 11. fputs("\n",fp); 12. } 13. printf("Finished"); 14. fclose(fp); 15. }</pre> |

| ผลลัพธ์ |
|---------|
| |

10.7 การอ่านแฟ้มข้อมูล

10.7.1 การอ่านแฟ้มข้อมูลด้วยคำสั่ง fscanf()

การเขียนลงแฟ้มข้อมูลใช้คำสั่ง fscanf() โดยรูปแบบการใช้คำสั่งคล้ายกับคำสั่ง scanf() ปกติ

| รูปแบบคำสั่งการอ่านไฟล์ด้วยคำสั่ง fscanf() |
|--|
| <code>fscanf(ตัวแปรพอยน์เตอร์, "ข้อมูลแสดงผล+การควบคุมการแสดงผล", &ตัวแปร);</code> |
| ตัวอย่าง |
| <ol style="list-style-type: none"> 1. FILE *fptr; 2. int number; 3. float price; 4. fptr = fopen("test.txt", "r"); 5. fscanf(fptr, "%d", &number); 6. fscanf(fptr, "%f", &price) |

ตัวอย่างที่ 1 การอ่านแฟ้มข้อมูลด้วยคำสั่ง fscanf()

| โปรแกรม | ตัวอย่าง |
|--|----------|
| <pre> 1. #include <stdio.h> 2. main() 3. { 4. int num; 5. FILE *fptr; 6. if ((fptr = fopen("input.txt","r")) == NULL) 7. { 8. printf("Error! opening file"); exit(1); 9. } 10. fscanf(fptr,"%d", &num); 11. printf("Value of n=%d", num); 12. fclose(fptr); 13. }</pre> | |

ตัวอย่างที่ 2 การอ่านแฟ้มข้อมูลด้วยคำสั่ง fscanf()

| โปรแกรม | ตัวอย่าง |
|---|----------|
| <pre> 1. #include <stdio.h> 2. main() 3. { 4. int num; 5. FILE *fptr; 6. if ((fptr = fopen("input.txt","r")) == NULL){ 7. printf("Error! opening file"); exit(1); 8. } 9. while(feof(fptr) ==0){ 10. fscanf(fptr,"%d", &num); 11. printf("Value of n=%d\n", num); 12. } 13. fclose(fptr); 14. }</pre> | |

10.7.3 การอ่านแฟ้มข้อมูลด้วยคำสั่ง fgetc()

เป็นคำสั่งที่ใช้อ่านอักขระครั้งละ 1 ตัวจากแฟ้มข้อมูล

| รูปแบบคำสั่งการอ่านไฟล์ด้วยคำสั่ง fgetc() |
|--|
| ตัวแปร = fgetc(ตัวแปรพอยน์เตอร์); |
| ตัวอย่าง |
| <pre> 1. FILE *fptr; 2. char ch; 3. fptr = fopen("test.txt", "r"); 4. ch = fgetc(fptr); </pre> |

ตัวอย่างการอ่านแฟ้มข้อมูลด้วยคำสั่ง fgetc()

| โปรแกรม |
|--|
| <pre> #include <stdio.h> 1. main() 2. { 3. FILE *fp; 4. char ch = 'y'; 5. int i; 6. fp = fopen("input-fputc.txt", "r"); 7. while(feof(fp) == 0) 8. { 9. ch = fgetc(fp); 10. printf("%c", ch); 11. } 12. fclose(fp); 13. } </pre> |

| ผลลัพธ์ |
|---------|
| |

10.7.3 การอ่านเพิ่มข้อมูลด้วยคำสั่ง fgets()

เป็นคำสั่งที่ใช้อ่านข้อความครั้งละ 1 บรรทัดจากเพิ่มข้อมูล

| รูปแบบคำสั่งการอ่านไฟล์ด้วยคำสั่ง fgets() |
|--|
| <code>fgets(ชื่อตัวแปร,ความยาวตัวอักษร, ตัวแปรพอยน์เตอร์);</code> |
| ตัวอย่าง |
| <pre> 1. FILE *fptr; 2. char ch[100]; 3. fptr = fopen("test.txt", "r"); 4. fgets(ch, 10, fptr); </pre> |

ตัวอย่างการอ่านแฟ้มข้อมูลด้วยคำสั่ง fgets()

| โปรแกรม |
|--|
| <pre> 1. #include <stdio.h> 2. main() 3. { 4. FILE *fp; 5. char ch ='y'; 6. int i; 7. fp = fopen("input-fputc.txt", "r"); 8. fgets(ch,10,fp); 9. printf("%s",ch); 10. fclose(fp); 11. }</pre> |
| ผลลัพธ์ |
| |

10.8. ฟังก์ชันที่ใช้เกี่ยวกับการประมวลผลแฟ้มข้อมูล

10.8.1 ฟังก์ชันremove()

ใช้สำหรับการลบแฟ้มที่ระบุออกจากพื้นที่ฮาร์ดดิสก์

รูปแบบ: remove(ชื่อแฟ้ม);

เช่น ต้องการลบแฟ้ม c:\\student.txt

```
remove(c:\\student.txt);
```

10.8.2 ฟังก์ชัน rewind()

ใช้สำหรับการกำหนดให้ตัวชี้เพิ่มข้อมูลกลับไปชี้ยังจุดแรกเพิ่มข้อมูล

รูปแบบ : rewind(ตัวชี้เพิ่ม

เช่น

```
rewind(ptrData);
```

10.8.3 ฟังก์ชัน fflush()

ใช้สำหรับการล้างค่าให้ตัวชี้เพิ่มข้อมูลไม่ให้มีข้อมูลใดอยู่ภายใน

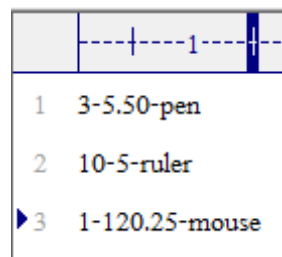
รูปแบบ : fflush(ตัวชี้เพิ่ม);

เช่น

```
fflush(ptrData);
```

คำถามท้ายบทที่ 10

1. จากการประกาศตัวแปร FILE *pointer_file; จงเขียนคำสั่งใช้พอยน์เตอร์ชื่อ pointer_file เพื่อคำสั่งเปิดไฟล์ชนิดไบนารีชื่อ roomLab10.bin แบบอ่าน โดยไฟล์ดังกล่าวอยู่ใน drive E และ โฟลเดอร์ชื่อ jirawan
2. จากการประกาศตัวแปร FILE *pointer_file; จงเขียนคำสั่งใช้พอยน์เตอร์ชื่อ pointer_file เพื่อคำสั่งเปิดไฟล์ ชื่อ roomLab10.txt แบบเขียนใหม่ โดยไฟล์ดังกล่าวอยู่ใน drive C และ โฟลเดอร์ชื่อ Jirawan
3. จงอธิบายความหมายคำสั่ง feof และ fclose ของการประมวลผลไฟล์
4. จงบอกความแตกต่างระหว่างคำสั่ง fputs และ puts
5. จงบอกความแตกต่างระหว่างคำสั่ง fprintf และ printf
6. ให้นักิตสร้างไฟล์ชื่อ data.txt ให้มีรูปแบบดังต่อไปนี้



และเขียนโปรแกรมคิดราคาของสินค้าพร้อมทั้งแสดงผลลัพธ์ตามตัวอย่างหน้าจอคอมพิวเตอร์

ด้านล่าง โดยกำหนดให้ใช้คำสั่ง fscanf เพื่ออ่านค่าจากไฟล์ data.txt เท่านั้น

```
Name = pen      Number = 3      Price = 5.50      Total = 16.50
Name = ruler    Number = 10     Price = 5.00      Total = 50.00
Name = mouse    Number = 1      Price = 120.25    Total = 120.25
```

7. จงเขียนโปรแกรมเขียนไฟล์นามสกุล txt เพื่อบันทึกข้อมูลส่วนตัวของนิสิต ประกอบด้วยรหัสนิสิต ชื่อ นามสกุล และเกรดเฉลี่ยนิสิต โดยรับค่าข้อมูลดังกล่าวผ่านหน้าจอคอมพิวเตอร์
8. จงเขียนโปรแกรมเขียนไฟล์นามสกุล txt เพื่อบันทึกข้อมูลส่วนตัวของเพื่อนนิสิตจำนวน 5 คน ประกอบด้วยรหัสนิสิต ชื่อ นามสกุล และเกรดเฉลี่ยนิสิต โดยรับค่าข้อมูลดังกล่าวผ่านหน้าจอคอมพิวเตอร์ โดยกำหนดให้ใช้คำสั่ง fputs เขียนข้อมูลชนิดสายอักขระ

9. ให้นิสิตรับค่าชื่อสินค้าและราคาจำนวน 3 ชนิด จากแป้นพิมพ์ จากนั้นให้นิสิตนำข้อมูลดังกล่าวมาเขียนลงในไฟล์ 2 ไฟล์ โดยกำหนดให้
- 1) ไฟล์ชื่อ product.txt เป็นไฟล์ที่เก็บชื่อสินค้า
 - 2) ไฟล์ชื่อ price.txt เป็นไฟล์ที่เก็บชื่อสินค้า โดยเก็บเป็นทศนิยม 2 ตำแหน่ง
 - 3) หน้าจอการรับและแสดงผลข้อมูลเป็นไปตามตัวอย่างที่กำหนด

```

Please input product name :
PaperA4
Please input product price :
100

*****
Please input product name :
Egg
Please input product price :
4.500

*****
Please input product name :
Milk
Please input product price :
15.5555

*****

```

10. จากโปรแกรมและผลการทำงานบนจอมอนิเตอร์ดังภาพด้านล่าง จงแสดงข้อมูลภายในไฟล์ emp.txt

```

Enter the name : Steve
Enter the salary : 1000
*****
Enter the name : Bill
Enter the salary : 2000
*****

```

| โปรแกรม | ข้อมูลภายในไฟล์ emp.txt |
|-----------------------------------|-------------------------|
| 1. #include <stdio.h> | |
| 2. main() | 1. |
| 3. { | 2. |
| 4. FILE *fptr; | 3. |
| 5. char name[30]; | 4. |
| 6. float salary; | 5. |
| 7. int i; | 6. |
| 8. fptr = fopen("emp.txt", "w+"); | 7. |
| 9. for(i=0;i<2;i++){ | 8. |
| 10. printf("Enter the name : "); | |

| | |
|--|--|
| <pre>11. scanf("%s", &name); 12. fprintf(fp, "Name= %s\n", name); 13. printf("Enter the salary : "); 14. scanf("%f", &salary); 15. fprintf(fp, "Salary= %.2f\n", salary); 16. printf("*****\n"); 17. } 18. fclose(fp); 19. }</pre> | |
|--|--|