文件与错误

C Reference

Text files

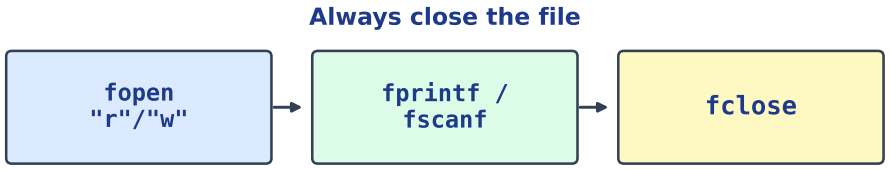
fopen 返回一个文件指针 (file pointer); 模式 (mode) 字符串说明要做什么 —"w" 写、"r" 读、"a" 追加。用 fprintf 写, 用 fscanf 读, 用完后一定要 fclose。

```c
#include <stdio.h>

int main(void) {
    FILE *out = fopen("scores.txt", "w");   // open for writing
    if (out == NULL) {
        perror("scores.txt");
        return 1;
    }
    fprintf(out, "Alice 80\nBob 95\n");
    fclose(out);

    FILE *in = fopen("scores.txt", "r");    // open for reading
    if (in == NULL) {
        perror("scores.txt");
        return 1;
    }
    char name[20];
    int score;
    while (fscanf(in, "%19s %d", name, &score) == 2) {   // %19s leaves
↪   room for '\0'
        printf("%s -> %d\n", name, score);
    }
    fclose(in);
    return 0;
}
```


exercises.lol exercises.com.cn



Exam tip: check fopen != NULL; mode "r" reads, "w" writes; fclose when done

fopen → read/write → fclose

Error handling with return codes

C 语言没有内置异常机制。函数可通过返回码报告结果；使用前请查阅该函数的规则。下例中 0 表示成功，-1 表示失败。fopen 则返回流指针，失败时返回 NULL。

```c
#include <stdio.h>

// returns 0 on success, -1 if the divisor is 0
int safe_divide(int a, int b, int *result) {
    if (b == 0) return -1;          // error code
    *result = a / b;
    return 0;                       // success
}

int main(void) {
    int r;
    if (safe_divide(10, 2, &r) == 0)
        printf("10 / 2 = %d\n", r);          // 10 / 2 = 5
    if (safe_divide(10, 0, &r) != 0)
        printf("cannot divide by zero\n");   // error reported
    return 0;
}
```

常见错误

- 读写前先检查 fopen 没有返回 NULL。
- 用完后一定要 fclose 文件。
- 从 main 返回非零值, 向调用者表示出错。