

文件与错误

C Reference

文本文件

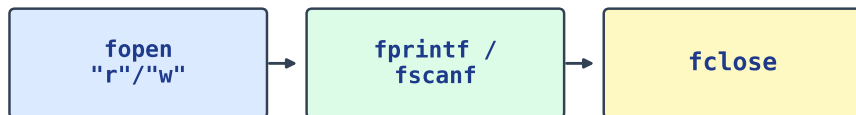
`fopen` 返回一个文件指针 (file pointer); 模式 (mode) 字符串说明要做什么 —— "w" 写、"r" 读、"a" 追加。用 `fprintf` 写, 用 `fscanf` 读, 用完后一定要 `fclose`。

```
#include <stdio.h>

int main(void) {
    FILE *out = fopen("scores.txt", "w");    // 打开以写入
    fprintf(out, "Alice 80\nBob 95\n");
    fclose(out);

    FILE *in = fopen("scores.txt", "r");    // 打开以读取
    char name[20];
    int score;
    while (fscanf(in, "%s %d", name, &score) == 2) {
        printf("%s -> %d\n", name, score);
    }
    fclose(in);
    return 0;
}
```

Always close the file



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

fopen → *read/write* → *fclose*

用返回码处理错误

C 没有异常机制。函数改用返回码 (return code) 来报告失败 —— 按惯例 0 表示成功, 非零表示错误。调用者在信任结果之前先检查这个码。(fopen 遵循同样的思路: 失败时返回 NULL。)

```
#include <stdio.h>

// 成功返回 0, 除数为 0 时返回 -1
int safe_divide(int a, int b, int *result) {
    if (b == 0) return -1;    // 错误码
    *result = a / b;
    return 0;    // 成功
}

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");    // 报告错误
    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。