Ввод-вывод

Т.И. Комаров

НИЯУ МИФИ

2023

Иерархия памяти I

Иерархия памяти эволюционирует для обеспечения минимального времени доступа и максимальной производительности

Характеристики памяти:

- Емкость
- Время доступа
- Производительность
- Стоимость на 1 байт

Иерархия памяти II

Тип памяти	Время доступа	Емкость	Управление
Регистры	1 цикл	1 Кбайт	Программа/ОС
Кэш L1	2-4 цикла	32 Кбайт	Аппаратное
Кэш L2	10 циклов	256 Кбайт	Аппаратное
Кэш L3	40 циклов	10 Мбайт	Аппаратное
Основная память	200 циклов	10 Гбайт	Программа/ОС
Внешняя память	>>>>>>	Разная	Программа/ОС

Файлы

Определения

Файловая система — порядок, определяющий способ организации, хранения и именования данных на носителях информации в компьютерах

Файл — именованная область данных на носителе информации

Операции с файлами:

1. Открытие файла
2. Закрытие файла
3. Чтение из файла
4. Запись в файл
5. Позиционирование в файле произвольного доступа

Файлы в UNIX-подобных ОС I

Один из принципов UNIX-подобных ОС

«Всё есть файл»

Весь ввод-вывод осуществляется путём чтения и записи файлов. Все периферийные устройства, а так же некоторые средства межпроцессного взаимодействия представлены в виде файлов, интерфейс взаимодействия с ними унифицирован

Файлы в UNIX-подобных ОС II

Идея

Системный вызов — обращение к ядру ОС с просьбой выполнить какое-либо действие. Любое взаимодействие с «внешним миром» требует обращения к ядру ОС

Ядро ОС предоставляет интерфейс системных вызовов, среди которых есть некоторые, предназначенные для работы с файлами:

- `int open(const char *pathname, int flags, mode_t mode)`
- `int creat(const char *pathname, mode_t mode)`
- `int close(int fd)`
- `ssize_t read(int fd, void *buf, size_t count)`
- `ssize_t write(int fd, const void *buf, size_t count)`
- `off_t lseek(int fd, off_t offset, int whence)`

Файлы в UNIX-подобных ОС III

Идея

Файловый дескриптор — неотрицательное целое число, которое идентифицирует открытый файл

Системный вызов `open()/creat()` возвращает программе файловый дескриптор, который используется во всех прочих системных вызовах для работы с файлами

Вся работа с файлом и информация о нём реализуется внутри ядра ОС, прикладная программа работает только с файловым дескриптором

Файлы в UNIX-подобных ОС IV

При запуске программы автоматически открываются три файла, которые соответствуют трем стандартным потокам и имеют дескрипторы 0, 1 и 2:

1. Стандартный поток ввода — `STDIN_FILENO`
2. Стандартный поток вывода — `STDOUT_FILENO`
3. Стандартный поток ошибок — `STDERR_FILENO`

Файлы в UNIX-подобных ОС V

Примеры

```c
#include <stdio.h>
#include <unistd.h>

// Copy stdin to stdout
int main() {
    char buf[BUFSIZ];
    int n;
    // All return values must be checked
    while ((n = read(0, buf, BUFSIZ)) > 0) {
        write(1, buf, n);
    }
    return 0;
}
```

Файлы в UNIX-подобных ОС VI

Важно

Интерфейс системных вызовов является специфичным для каждого ядра ОС

Даже в UNIX-подобных ОС интерфейс системных вызовов может значительно различаться

Поэтому для разработки прикладного ПО, которое должно работать в различных ОС, обычно используют функции из стандартной или других библиотек, являющиеся кроссплатформенной обёрткой над системными вызовами

Работа с файлами средствами стандартной библиотеки (открытие файла) I

Доступные функции:

1. `FILE *fopen(const char *restrict pathname, const char *restrict mode)`
2. `FILE *fdopen(int fd, const char *mode)`
3. `FILE *freopen(const char *restrict pathname, const char *restrict mode, FILE *restrict stream)`

Работа с файлами средствами стандартной библиотеки (открытие файла) II

Параметр функций `mode` может принимать следующие значения:

- `r` — открыть файл для чтения, поток позиционируется на начало файла.
- `r+` — открыть файл для чтения и записи, поток позиционируется на начало файла.
- `w` — создать файл для записи или очистить существующий файл («урезать» до нулевой длины), поток позиционируется на начало файла.
- `w+` — создать файл для чтения и записи или очистить существующий файл («урезать» до нулевой длины), поток позиционируется на начало файла.

Работа с файлами средствами стандартной библиотеки (открытие файла) III

- a — открыть файл для «дописывания» (записи в конец файла) или создать его в случае отсутствия, поток позиционируется на конец файла.
- a+ — открыть файл для чтения и «дописывания» (записи в конец файла) или создать его в случае отсутствия, запись осуществляется только в конец файла, начальное позицирование потока для чтения не регламентировано стандартом.

Дополнительно параметр `mode` может содержать в конце символы `b` или `t` (игнорируются в POSIX-совместимых ОС).

Работа с файлами средствами стандартной библиотеки (закрытие файла)

Доступная функция:

- `int fclose(FILE *stream)`