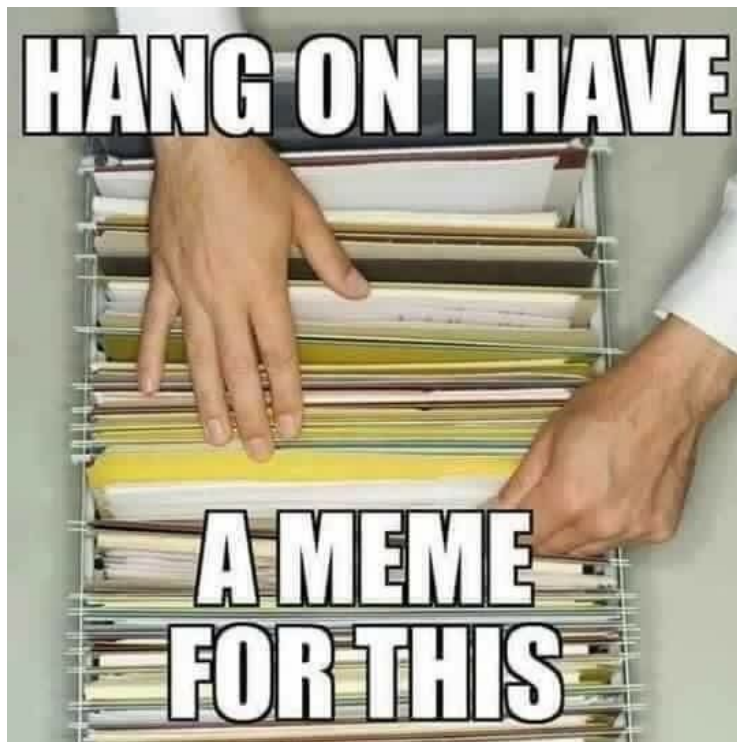


Семинар 12 по УП

Файлове



Какво е файл?

- Много ама много неформална дефиниция: Масив от байтове, който се съхранява на хард диска
- Правилно чухте байтове. Всичко накрая става такова и от програмата зависи как ще го интерпретира.

Защо?

- Защото може би искаме да запазим данните и след затваряне на програмата.
- Може би искаме да променяме съществуващи файлове.
- Рамта е супер памет, но е енергозависима.

Предимства

- Енергонезависимост
- По-голям капацитет (Не ми се вярва рамта да ви е повече от хардовете)

Недостатъци

- Относително бавни за четене и писане
- В началото са трудни за разбиране

Видове файлове

- Текстови файлове
- Двоични файлове

Текстови файлове

- Това са файлове, които ние можем да четем и с просто око. От тях се четът по-лесно стандартни типове.

Двоични файлове

- Можем да “набутаме” всичко в този файл в суров вид. Не ни интересува как ще изглежда. Не добър вид за четене от човек (никак даже). Можем да “налеем” информацията от него в каквато си искаме структура и в какъвто искаме тип.

Как се работи с файлове?

- Нужна ни е библиотеката `fstream`
- Има 3 вида потоци `ifstream`, `ofstream` и `fstream`.
- Има флагове `ios::in`, `ios::out`, `ios::binary`, `ios::trunc`, `ios::app`, `ios::ate`
- След отваряне на файл се създават указатели, като ако е отворен само за четене се създава само такъв за четене, аналогично само за писане се създава указател само за писане и ако е отворен и за четене и за писане се създават и двата указателя.

Как се работи с файлове?

- Отваряме файла с метод `open(име, флагове)`
- Затваряме с метод `close()`;
- Местим указател за четене с метод `seekg(брой позиции, локация) (seek get)`
- Местим указател за писане с метод `seekp(брой позиции, локация) (seek put)`

Споменах думата локация

- Споко не са географски координати
- `ios::beg` – началото на файла
- `ios::cur` – текуща позиция
- `ios::end` – край на файла

А как да четем и пишем?

- Тук идва кефа. Както използвате cin и cout, само че с името на потока, който отворихте.

```
std::fstream fstr("arr.txt", std::ios::in | std::ios::out);  
fstr << "Hello";
```

```
char ar;  
fstr >> ar;
```

Малко въпросчета

- Как ще взема предпоследния символ на файл?
- Как ще взема втория символ на файл?
- Как ще изместя указателя с една позиция напред?
- А с една назад?

Какво става при двоични файлове

- Scary Things Alert
- Предупреждение номер 2
- Добре де щом толкова искате (а имате ли избор?)
- При отварянето се “вдига” флаг `ios::binary`

Ии сега

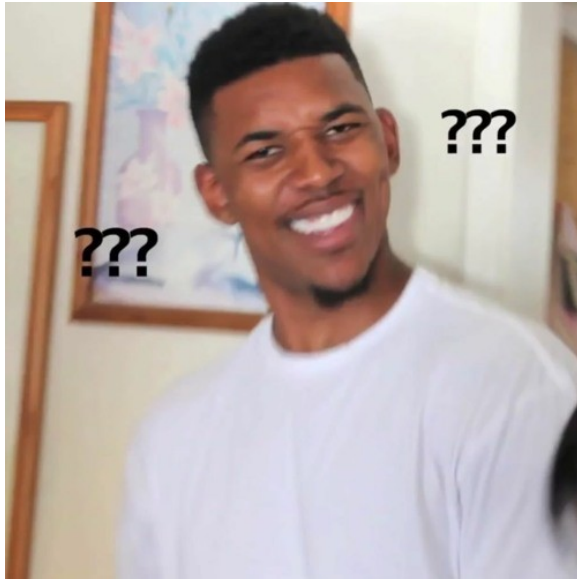
- Когато работим с двоичен файл ние четем и пишем по друг начин. Споменах че това е масив от байтове. Идеята на цялото нещо е просто да кажем искаме да прочетем някакво количество байтове и да го запишем в тази променлива. Или искаме да запишем толкова байтове във файла от паметта на тази променлива.

Казах ви, че е мътна и кървава.

```
std::ifstream fstr("arr.txt", std::ios::in | std::ios::binary);
char arr[6];
fstr.read(arr, sizeof(arr));
std::cout << arr;
```

```
std::ofstream fs("arr2.txt", std::ios::out | std::ios::binary);
Point a {5 , 6};
fs.write((char*)&a, sizeof(a));
fs.close();
Point b;
std::ifstream fstr("arr2.txt", std::ios::in | std::ios::binary);
fstr.read((char*)&b, sizeof(b));
std::cout << b.x << " " << b.y << std::endl;
```


WTF is this



- Реално само синтаксисът е гаден. Като се замислите това прави точно каквото ви казах. Взема адреса в паметта на нещо. Каства го към `char*`, защото файлът е масив от байтове и то чете байтове. А в `c++` байтът е `char`. Ии после задава колко байта да се пишат / четат.

Задачи!

- Напишете програма, която въвежда от двоичен файл масив от точки и записва прочетения масив в текстов файл.
- Напишете програма, която поддържа във файл информация за потребителите и има следните команди:
 - register – приема потребителско име, парола, име, фамилия и година на раждане
 - login – приема потребителско име и парола и връща данните на потребителя
 - removeUser – приема потребителско име и премахва потребителя от файла
- Напишете програма, която въвежда масив числа от двоичен файл, сортира го и го извежда в текстов.