**Тема**: Файловая система.

**Что нужно знать**:

* данные на дисках хранятся в виде файлов (наборов данных, имеющих имя)
* чтобы было удобнее разбираться с множеством файлов, их объединяют в каталоги (в *Windows* каталоги называются «папками»)
* каталог можно воспринимать как контейнер, в котором размещаются файлы и другие каталоги, которые называются подкаталогами или *вложенными* каталогами (они находятся внутри другого каталога, вложены в него)
* каталоги организованы в многоуровневую (иерархическую) структуру, которая называется «деревом каталогов»
* главный каталог диска (который пользователь видит, «открыв» диск, например, в Проводнике *Windows* или аналогичной программе) называется *корневым* каталогом или «корнем» диска, он обозначается буквой логического диска, за которой следует двоеточие и знак «\» (обратный слэш[[1]](#footnote-1)); например, **A:\** – это обозначение корневого каталога диска А
* каждый каталог (кроме корневого) имеет (один единственный!) «родительский» каталог – этот тот каталог, внутри которого находится данный каталог
* полный адрес каталога – это перечисление каталогов, в которые нужно войти, чтобы попасть в этот каталог (начиная с корневого каталога диска); например  
   **С:\USER\BIN\SCHOOL**
* полный адрес файла состоит из адреса каталога, в котором он находится, символа «\» и имени файла, например  
   **С:\USER\BIN\SCHOOL\Вася.txt**
* маска служит для обозначения (выделения) группы файлов, имена которых имеют общие свойства, например, общее расширение
* в масках, кроме «обычных» символов (допустимых в именах файлов) используются два специальных символа: звездочка «\*» и знак вопроса «?»;
* звездочка «\*» обозначает любой количество любых символов, в том числе, может обозначать пустую последовательность;
* знак вопроса «?» обозначает ровно один любой символ
* при выводе списка имен файлов они могут быть отсортированы по имени, типу (расширению), дате последнего изменения, размеру; это не меняет их размещения на диске;
* если установлена сортировка по имени или типу, сравнение идет по кодам символов, входящих в имя или в расширение

### Пример задания:

*Определите, какое из указанных имен файлов удовлетворяет маске:* **?hel\*lo.c?\***

1) **hello.c** 2) **hello.cpp** 3) **hhelolo.cpp** 4) **hhelolo.c**

**Решение:**

1. будем проверять соответствие файлов маске по частям, записывая результаты в таблицу
2. начнем с первой части маски, «**?hel**»; эта часть означает, что перед сочетанием «**hel**» в начале имени стоит один любой символ;
3. сразу видим, что первые два имени не подходят (начинаются прямо с «**hel**», без стартового символа), отмечаем их крестиком в таблице и больше не рассматриваем:

|  |  |
| --- | --- |
|  | **?hel** |
| **hello.cp** | **×** |
| **hello.cpp** | **×** |
| **hhelolo.cpp** | **hhelolo.cpp** |
| **hhelolo.c** | **hhelolo.c** |

желтым и фиолетовым маркером в таблице выделены соответствующие части маски и имен файлов (где есть совпадение)

1. для двух последних имен проверяем второй блок маски: после «**hel**» должна быть цепочка «**lo.c**», или вплотную (и это возможно!) или через произвольную «вставку», на которую указывает звездочка в маске; видим, что оба имени прошли проверку:

|  |  |  |
| --- | --- | --- |
|  | **?hel** | **?hel\*lo.c** |
| **hello.cp** | **×** |  |
| **hello.cpp** | **×** |  |
| **hhelolo.cpp** | **hhelolo.cpp** | **hhelolo.cpp** |
| **hhelolo.c** | **hhelolo.c** | **hhelolo.c** |

1. последняя часть маски, «**?\***», означает, что после «**lo.c**» должен стоять по крайне мере один любой символ (на это указывает знак «?»); проверяя это правило, обнаруживаем, что для последнего имени, «**hhelolo.c**», маска не подходит, поскольку после «**lo.c**» ни одного символа нет:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **?hel** | **?hel\*lo.c** | **?hel\*lo.c?\*** |
| **hello.cp** | **×** |  |  |
| **hello.cpp** | **×** |  |  |
| **hhelolo.cpp** | **hhelolo.cpp** | **hhelolo.cpp** | **hhelolo.cpp** |
| **hhelolo.c** | **hhelolo.c** | **hhelolo.c** | **×** |

1. таким образом, правильный ответ – 3.

|  |
| --- |
| **Возможные ловушки и проблемы**:   * + можно забыть, что звездочка «\*» может соответствовать и пустой последовательности; например, в рассмотренной задаче имя «**hhelolo.cp**» также соответствует маске   + можно забыть, что знак «?» НЕ может соответствовать пустой последовательности, а заменяет ровно 1 символ |

### Еще пример задания:

*Перемещаясь из одного каталога в другой, пользователь последовательно посетил каталоги* **DOC***,* **USER***,* **SCHOOL***,* **A:\***,* **LETTER***,* **INBOX***. При каждом перемещении пользователь либо спускался в каталог на уровень ниже, либо поднимался на уровень выше. Каково полное имя каталога, из которого начал перемещение пользователь?*

1) **A:\DOC**

2) **A:\LETTER\INBOX**

3) **А:\SCHOOL\USER\DOC**

4) **А:\DOC\USER\SCHOOL**

**Решение:**

1. сразу отметим, что здесь предполагается, что пользователь не переходил на другой диск;
2. в задачах, где нужно определить полный адрес файла или каталога, нужно начинать с поиска имени диска, в данном случае каталог находится на диске **A:\** (так обозначается корневой каталог)
3. поскольку в списке посещенных каталогов перед **A:\** стоит **SCHOOL**, пользователь мог попасть в корень диска **A:\** только через каталог **SCHOOL**, поэтому адрес стартового каталога начинается с **A:\SCHOOL**
4. так как среди предложенных вариантов только один удовлетворяет этому условию, можно остановиться (правильный ответ – 3), однако, давайте все же доведем процедуру до конца, это позволит выяснить некоторые интересные моменты, которые не сработали здесь, но могут сработать в других задачах
5. как пользователь попал в каталог **SCHOOL**? по условию он мог как подниматься, так и спускаться по дереву каталогов
6. явно он не мог спуститься из **USER** в **SCHOOL**, потому что **SCHOOL** находится в корневом каталоге и каждый каталог имеет только одного «родителя»; значит, пользователь поднялся из **USER** в **SCHOOL**, и начальная часть полного адреса **А:\SCHOOL\USER**
7. аналогично можно доказать, что пользователь поднялся в каталог **USER** из каталога **DOC**
8. таким образом, мы доказали, что правильный ответ – 3.

|  |
| --- |
| **Возможные ловушки и проблемы**:   * + можно забыть, что названия каталогов в полном имени перечисляются в порядке входа в них, начиная от корневого; пользователь *выходил* из каталогов, поднимаясь к корню диска **A:\**, поэтому проходил каталоги в обратном порядке (на эту ошибку рассчитан неверный ответ **А:\DOC\USER\SCHOOL**)   + можно перепутать каталог, из которого вышел пользователь, и каталог, где он в конечном счете оказался (на эту ошибку рассчитан неверный ответ **А:\LETTER\INBOX**)   + в условии есть лишние данные, которые только запутывают дело; например, имена каталогов **LETTER***,* **INBOX** никак не влияют на ответ, потому что пользователь попал в них уже *после* выхода в корневой каталог диска **A:\**, то есть, пройдя весь путь исходного каталога в обратном порядке |

### Еще пример задания:

*Каталог содержит файлы с именами*

а) **q.c**

б) **qq.cpp**

в) **qq.c**

г) **q1.c1**

д) **qaa.cmd**

е) **q12.cpp**

*Определите, в каком порядке будут показаны файлы, если выбрана сортировка по типу (по возрастанию).*

1) **авгдбе** 2) **авгдеб** 3) **абвгде** 4) **авдбег**

**Решение:**

1. при сортировке по типу сравниваются расширения имен файлов
2. при сравнении используют коды символов
3. отсутствие символа (когда расширение закончилось) считается «меньше» любого символа, то есть, файл с расширением **.с** будет находиться в списке выше, чем файлы с расширениями **.c1** и **.cmd**
4. коды цифр размещаются в таблице символов раньше, чем коды букв, то есть, файл с расширением **.с1** будет находиться в списке выше, чем файл с расширением **.сmd**
5. теперь можно распределить имена файлов по расширениям

а) **q.c**

в) **qq.c**

г) **q1.c1**

д) **qaa.cmd**

б) **qq.cpp**

е) **q12.cpp**

1. осталась еще одна проблема – решить, что делать, если расширения совпадают; в этом случае в большинстве программ для определенности используется дополнительная сортировка по имени, поэтому файл с именем **q12.cpp** будет стоять в списке выше, чем файл **qq.cpp** (код цифры ‘1' меньше, чем код буквы ‘q’)
2. в итоге получаем

а) **q.c**

в) **qq.c**

г) **q1.c1**

д) **qaa.cmd**

е) **q12.cpp**

б) **qq.cpp**

1. таким образом, мы доказали, что правильный ответ – 2.

|  |
| --- |
| **Возможные ловушки и проблемы**:   * + можно забыть правильно расставить имена файлов с одинаковыми расширениями (неверный ответ 1)   + нельзя сравнивать числовые значения: например, интуитивно кажется, что файл с расширением **.c10** «больше», чем файл с расширением **.c2**, однако это неверно, потому что код цифры '2' больше, чем код цифры '1’; поэтому файл с расширением **.c10** будет стоять в списке выше файла с расширением **.c2** (при сортировке по типу в порядке возрастания)   + можно забыть, что отсутствие кода (имя или расширение закончилось) «меньше» любого кода   + можно забыть, что коды цифр меньше, чем коды букв   + очень легко по невнимательности выбрать не тот ответ |

1. Для разделения имен каталогов в адресе в разных операционных системах применяют прямой слэш «/» или обратный слэш «\». В системе *Windows*, которая наиболее распространена в России, стандартным разделителем считается «\», именно такой знак чаще всего используется в задачах ЕГЭ. [↑](#footnote-ref-1)