

# Графические Системы. Часть II

## Лекция № 4

**Программирование графического  
пользовательского интерфейса  
средствами X-WINDOW.**

**ИПВУ. Управляющие объекты – widget'ы и  
их ресурсы. Иерархия widget'ов**

# Основы программирования в системе X Window System.

## РЕСУРСЫ X-Клиентов

Итак, одна из базовых концепций **Window System**, придающая системе большую гибкость, **это концепция РЕСУРСОВ**

Как уже говорилось выше - не следует заводить пару **widget'ов** "красная кнопка" и "зеленая кнопка", если можно завести один **widget** "кнопка" с параметром "цвет" (тем более, в дальнейшем разработчику понадобится еще и "желтая кнопка"). Поэтому **widget** должен быть достаточно универсальным: у него должно быть много параметров, списки которых отличаются от **widget'a** к **widget'y**, и, в зависимости от параметров, объект - **widget**

должен **РИСОВАТЬСЯ** и **реагировать** на события по-разному.

Такие (поименованные) параметры или атрибуты **widget'ов** называются **ресурсами**. Ресурсами **widget'ов** могут быть, например, цвет фона его окна, шрифт выводимого текста, цвет границы окна; механизм ресурсов может быть использован для определения модели поведения **widget'a** в зависимости от происходящих событий.

Главная проблема - как задавать и изменять **ресурсы widget'ов**? Не умозрительно, а руками, в программе? Различные параметры **widget'a** разбросаны по нескольким (иногда десяткам) вложенных друг в друга структур. Не помнить же наизусть названия всех этих структур?

Разработчики **X Window** предложили способ разрешения данной проблемы. Это **Механизм конфигурирования ресурсов**.



# Основы программирования в системе X Window System.

## РЕСУРСЫ X-Клиентов

Ресурсы разных типов можно конвертировать друг в друга. В частности, почти все типы допускают конвертирование в вид текстовой строки и обратно.

Например, тип "Цвет" естественно задавать либо в виде трех чисел (формат RGB), либо в виде названия этого цвета. Нетрудно сформировать (и это сделано в **X Window**) таблицу цветов: название и числа RGB. Пользователю удобнее работать с этими названиями, куда более содержательными, чем три числа. *[Существуют, однако, типы, не поддающиеся осмысленному конвертированию в вид текстовой строки, в частности, это внутренние параметры, связанные с расположением объектов в оперативной памяти. Все нижеследующее на них не распространяется].*

Если ресурсы почти всех типов можно конвертировать из вида текстовой строки, значит, можно завести текстовый файл со значениями каких-либо ресурсов. Файлы эти можно менять, не затрагивая исходный код приложения.

Будем называть их **конфигурационные файлы**.

Конфигурационные файлы состоят из директив задания значения ресурсов. При инициализации (**старте X-сессии**) эти директивы подгружаются в небольшую базу данных, поддерживаемую **XToolkit**. **Rdb** – это ресурсная база данных, используемая **X-Сервером** для предоставления окнам X-Клиентам различных характеристик для своего воспроизведения.

Далее, при создании каждого **widget'a**, наступает фаза переустановки значений ресурсов в соответствии с конфигурационными файлами. На этой фазе **XToolkit** опрашивает свою базу по каждому ресурсу в отдельности и из всех директив, которые могут относиться к данному ресурсу, выбирает максимально "специальную" директиву, наиболее подробно соответствующую этому ресурсу. Правила выбора этой "самой специальной" директивы достаточно сложны, но интуитивно понятны.

# Основы программирования в системе X Window System.

## РЕСУРСЫ X-Клиентов

Формат каждой директивы довольно прост:

### **путь\_к\_ресурсу: значение ресурса**

Что же такое путь к ресурсу? Ресурс - это часть **widget'a**. **Widget**, как правило, входит в состав другого **widget'a**. Тот в свою очередь... Но рано или поздно эта цепочка заканчивается головным **widget'ом** приложения, т.е **X-Клиента**. Остается теперь вспомнить, что у каждого **ресурса** и у каждого **widget'a** есть целых два имени: **имя объекта и имя класса**. Путь к ресурсу состоит из последовательности этих имен, разделенных точками.

Иными словами, в **X Window** файл ресурсов есть обычный текстовый файл, каждая строка которого задает тот или иной параметр (ресурс) программы. (При этом предполагается, что программу "населяют" именованные объекты – **widget'ы**, вернее их совокупности, связанные в некоторую иерархию - **Widget-tree**).

Общий вид строки следующий:

```
<имя программы>.<подобъект1>.<подобъект2>. . .  
                                <подобъектN>.<имя ресурса>: <значение ресурса>
```

Подобная строка задает значение ресурса для подобъектов иерархии объектов программы. Например, запись



# Основы программирования в системе X Window System.

## РЕСУРСЫ X-КЛИЕНТОВ

```
myprog.dialogwnd.background: Red
```

говорит, что в программе **myprog** у объекта- **widget'a** с именем **dialogWindow** параметр **background** (цвет фона) имеет значение **red** (красный цвет).

Вместо имен объектов- **widget'ов** могут указываться их классы. Обычно класс имеет то же самое имя, что и **widget**, но начинается с заглавной буквы, например,

```
Myprog.dialogwnd.Background: Red
```

Часть **widget'ов** или классов в левой части строки, задающей ресурс, может заменяться символом **'\***, например, строка

```
myprog*background: Red
```

указывает, что для всех объектов программы **myprog** ресурс **background** имеет значение **Red**.

Связка с помощью символа **':** имеет бОльший приоритет, чем связка с помощью **'\***. Так, если в файле, задающем ресурсы, есть две строки

```
myprog*background: Red  
myprog.dialogwnd.background: Green
```

то все объекты программы будут иметь ресурс **background** равный **Red**, кроме объекта **dialogwnd**, для которого этот параметр - **Green**.

# Основы программирования в системе X Window System.

## Создание базы данных ресурсов X-Клиентов

Загрузка ресурсов производится специальными инициализационными процедурами, которые конструируют базу данных ресурсов **X-сервера** для воспроизведения **Window-tree X-Клиента** из различных ресурсных файлов, опций командной строки вызова и других источников.

Для доступа к данным **X Window** предоставляет набор функций, которые совокупно называются **менеджер ресурсов (Resource Manager)**, и специальную программу **xrdb**, которая позволяет считать любой ресурсный файл и включить его в общую **таблицу ресурсов сервера**. Последняя называется **базой данных ресурсов сервера (resource data base)**, и представляет собой область памяти, ассоциированную со свойством **(property) XA\_RESOURCE\_MANAGER** корневого окна экрана дисплея (\*).

### Механизм свойств



# Основы программирования в системе X Window System.

## Подробнее – устройство X Window System. (\*)Свойства

Разработчики X нуждались в способе позволить клиентам обмениваться произвольными данными друг с другом, для этого используется механизм свойств (**properties**). **Свойство** - это **именованный набор данных**, связанный с некоторым объектом (например, окном), и доступный всем X-клиентам. Свойства используются X-клиентами, чтобы хранить информацию, которая может понадобиться другим X-клиентам, или если необходимо знать информацию используемую другими X-клиентами.

Схожесть с моделью "**ресурс-идентификатор**" очевидна, но и различия весьма существенны - если "**ресурсы**" можно потенциально разделять между программами-клиентами, работающими с одним X-сервером, то "**свойства**" являются общедоступными (для клиентов одного сервера). В этом же правиле можно заметить и дополнительный нюанс - атомы, в отличие от идентификаторов ресурсов, уникальны только для одного окна - ведь одно "свойство" может быть присуще разным окнам.

Каждое свойство имеет название и уникальный компактный идентификатор - **атом**. Обычно имена свойств записываются большими буквами, например: **MY\_PROPERTY**. Атомы используются для доступа к содержимому свойств с тем, чтобы уменьшить количество информации, пересылаемой по сети между клиентами и X сервером.

В X предусмотрен набор процедур, позволяющих перевести им свойства в уникальный атом, и, наоборот, по атому получить необходимые данные.

Некоторые свойства и соответствующие им атомы являются предопределенными и создаются в момент инициализации сервера. Этим атомам соответствуют символические константы, определенные в файлах-заголовках библиотеки **Xlib**.

Эти константы начинаются с префикса **XA\_**, например **XA\_WM\_HINTS**.

Ни к "содержимому" свойств, ни к их именам X Window не предъявляет фактически никаких особых требований, но существует фиксированное множество имен и соглашения о содержимом свойств, которые необходимы для нормального функционирования любого приложения в среде X Window. Эти соглашения об именах/содержании свойств с предопределенными атомами называются **ICCCM - "Соглашения о взаимодействии между программами-клиентами"**.

# Основы программирования в системе X Window System.

## Создание базы данных ресурсов X-Клиентов

Рассмотрим последовательность шагов по поиску и загрузке ресурсов  
в **rdb X-сервера**.

**Шаг № 1.** Загрузка ресурсов из командной строки старта **X-Клиента**.  
В **X- Window System** поддерживается стандартное множество опций –  
конфигурационных ресурсов **X-Клиента**, комбинации которых могут  
указываться в командной строке при запуске программы **X-Клиента**.

**Эти ресурсы перечислены в таблице:**





# Основы программирования в системе X Window System.

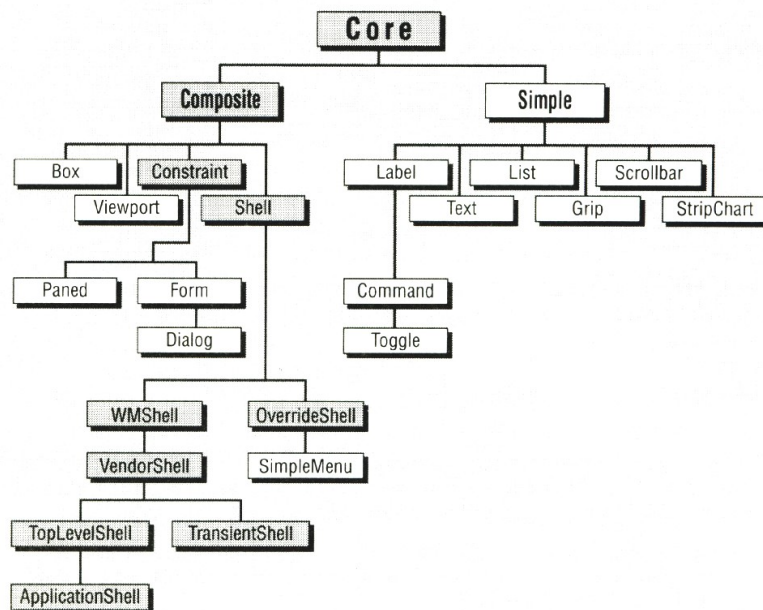
## Создание базы данных ресурсов X-Клиентов. Шаг № 1

Опции	Имя ресурса	Тип	Описание и примеры
-bg -background	*background	String	Цвет фона -bg blue
-bd -bordercolor	*borderColor	String	Цвет границы -bd black
-bw -borderwidth	.borderWidth	Integer	Ширина края окна в пикселах -bw 2
-display	.display	String	Сетевой ресурс для связи с сервером – дисплей -d CPUB:0.0
-fg -foreground	*foreground	String	Цвет «передника» -fg red
-fn -font	*font	String	Имя шрифта -fn 9x15
-geometry	.geometry	String	Размер и положение окна: -geometry 171x252
-iconic	.iconic	None	Иконизация окна – если значение «on»
-name	.name	String	Имя X-Клиента: -name VASJA
-rv -reverse	*reverseVideo	None	Инверсия видимого изображения – если значение «on»
+rv	*reverseVideo	None	Возврат к нормальному изображению
-title	.title	String	Заголовок окна: -title WINVASJA
-xrm	Значение аргумента	String	Ресурс и его значение задаются аргументом опции -xrm “*height:100”

# Основы программирования в системе X Window System.

## Создание базы данных ресурсов X-Клиентов

В данной таблице точка, стоящая перед именем ресурса, означает, что опции могут быть использованы для установки ресурса только в **widget'e**, принадлежащем классу **TopLevelShell** или его подклассу. В свою очередь, «\*» означает, что ресурс может быть установлен для произвольного widget'a **X-Клиента**.



Данный способ задания ресурсов – самый приоритетный с точки зрения управления процессом их загрузки в **rdb**.

**Шаг № 2.** Загружается файл, на который указывает системная переменная среды **XENVIRONMENT** (если таковая вообще задана). Данная переменная содержит полный путь к файлу, включая и его имя. Если же переменная **XENVIRONMENT** не установлена, то **rm** будет пытаться подгрузить файл «**.Xdefaults-*hostname***», который находится в домашней (**home**) директории пользователя. Здесь **<hostname>** - это имя компьютера, на котором происходит запуск **X-Клиента**



# Основы программирования в системе X Window System.

## Создание базы данных ресурсов X-Клиентов

**Шаг № 3.** Если «корневое» окно (**root window**) экрана имеет ресурсы, загруженные в «свойство» **XA\_RESOURCE\_MANAGER (RESOURCE\_MANAGER property)** программой **xrdb**, то они также добавляются в базу данных ресурсов **X-Клиента**. Если же «корневое» окно не имеет такого свойства, **rm** будет пытаться подгрузить данные, находящиеся в файле **«.Xdefaults»**, который находится в домашней (**home**) директории пользователя.

**Шаг № 4.** Различные варианты загрузки ресурсных файлов класса **X-Клиента**. Это может быть осуществлено через переменные окружения **XAPPLRESDIR**, **APPLRESDIR** или **XUSERFILESEARCHPATH** и др., которые позволяют задать путь к файлу ресурсов, например **«<\$ XAPPLRESDIR/<classname>»**. Здесь **<classname>** - имя класса **X-Клиента**. Если переменные не установлены, **rm** будет пытаться подгрузить данные, находящиеся в файле **«.Xdefaults»**, который находится в домашней (**home**) директории пользователя.

**Шаг № 5.** Rm осуществляет поиск файла:

**«usr/lib/X11/app-defaults/<classname>»**

Если этот файл существует то он загружается в базу данных ресурсов.

**Шаг № 6.** Если значения каких-либо ресурсов остались неустановленными, рассматриваются параметры, переданные с помощью определенных аргументов инициализационных процедур. Значения, переданные через параметры, представляются в виде массива указателей на строки. Каждая строка имеет вид:

**«<описание ресурса>: <значение ресурса>»**

# Основы программирования в системе X Window System.

## Widget'ы и их ресурсы. Формализация

Следует обратить особое внимание на то обстоятельство, что каждый ресурс имеет свой тип. Тип определяет, в скольких байтах оперативной памяти размещается ресурс и как трактовать их содержимое. **XToolkit** поддерживает около 40 встроенных типов ресурсов. Но X-программист может свободно завести новые типы.

### Для задания значения ресурсов разработан механизм преобразования типов ресурсов.

По аналогии с **widget'ами** для каждого ресурса должны быть определены имя типа ресурса; имя класса ресурса; имя ресурса;

Дополнительно должны быть известны привязка к месту ресурса в структуре данных **widget'a** (или класса **widget'ов**), а также значение ресурса по умолчанию.

### Таким образом можно подытожить :

В X Window System, в концепции ИПВУ (**XToolkit**), реализованы механизмы по работе с **widget'ами** :

- несколько "самых базовых" **widget'ов** с реализацией методов производных **widget'ов** с использованием методов базовых **widget'ов**;
- процедуры поддержки функционирования иерархии **widget'ов**;
- процедуры опроса и изменения ресурсов;
- механизмы управления потоком событий и реакции **widget'ов** на эти события;

# Основы программирования в системе X Window System.

## Widget'ы и их ресурсы. Преимущества

Введение понятия ресурсов снимает, как минимум, две проблемы.

Во-первых, это проблема адресации параметров (ресурсов) **widget'ов**, а значит доступа к ним, за счет механизма конфигурирования .

Во-вторых - унификация работы с ресурсами **X-сервера**; действительно, достаточно написать конвертор из типа ресурсов "текстовая строка" в тип ресурсов "шрифт", и работа со шрифтами станет гораздо удобнее.

**При более внимательном рассмотрении оказывается, что введенная схема предлагает универсальное решение еще одной проблемы - конфигурирования приложений под конкретного пользователя.**



По своему назначению ресурсы в **X-Window** очень похожи на то, что обозначается тем же термином "ресурсы" в **MS Windows**. Но вся идеология работы с ресурсами в **X** в корне отличается.

В частности, в **MS Windows** ресурсы являются частью бинарного исполняемого файла (например, **winword.exe**), и могут указываться или при компиляции программы, или меняться при помощи специальных редакторов, которые позволяют модифицировать бинарный файл (что, вообще говоря, является работой для квалифицированного программиста, и зачастую противоречит законам об авторских правах).

В **X-Window** же ресурсы существуют отдельно от исполняемого кода программы, в виде текстовых файлов, и могут свободно меняться или в этих файлах, или даже при помощи ключей в командной строке.

Таким образом, если в **MS Windows**, к примеру, перевод всех сообщений некоей программы на русский язык является "хакерской" задачей, то в **X-Window** подобное действие -- вполне стандартно, доступно любому пользователю и описано в документации.