

# Ресурсные файлы в X Window System

Многие программы имеют различные заранее подготавливаемые данные, которые в терминах X называются - ресурсами. Это могут быть цвета окон приложения, строки сообщений пользователю и т.д.

Как правило, программисты создают ресурсы каждый по-своему. В X Window сделана попытка унифицировать этот процесс.

# Формат файла ресурсов

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, связанные в некоторую иерархию). Общий вид строки следующий:

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

# Пример строки задания ресурса

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

*myprog.dialogwnd.background: Red*

говорит, что в программе ProgramName у объекта с именем DialogWindow параметр Background (цвет фона) имеет значение Red (красный цвет).

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

*Myprog.dialogwnd.Background: Red*

# Пример строки задания ресурса

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

```
myprog*background: Red
```

символ '\*' указывает, что для всех этих 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 есть Green.

# Доступ к ресурсам программ

Пусть ресурсный файл подготовлен. 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, и представляет собой область памяти, ассоциированную со свойством (property) XA\_RESOURCE\_MANAGER корневого окна экрана дисплея.

# Доступ к ресурсам программ

Наиболее простой является функция `XGetDefault()`. Она получает имя программы, имя ресурса и определяет значение последнего. При этом она последовательно совершает следующие шаги:

- сначала ресурс ищется в базе данных сервера (в свойстве `XA_RESOURCE_MANAGER`);
- если он не найден, то значение ресурса определяется по файлу `".Xdefaults"`, который ищется в домашней (`home`) директории пользователя;
- если задана переменная среды `XENVIRONMENT`, то ресурс ищется в файле, на который указывает эта переменная.

# Доступ к ресурсам программ

Если ресурс одновременно встречается в ".Xdefaults" и файле, определяемом XENVIRONMENT, то берется последнее значение.

В примере, приводимом ниже, используется XGetDefault(), чтобы получить строку, которую надо напечатать в окне программы. Предполагается, что имя программы - "example", а строка - ресурс с именем "MyResource", т.е. в файле ".Xdefaults" должна быть помещена, например, следующая запись:

```
..... example.MyResource : First,  
Example! .....
```

# Фрагмент программы, выполняющий чтение из файла ресурсов

Будет выглядеть следующим образом:

```
Display *display; GC gc;
Window window; char *string; . . . . .
/* Устанавливаем связь с сервером, получаем номер
экрana */
/* События, обрабатываемые программой */
XSelectInput (display, window, ExposureMask | KeyPressMask);
/* Получаем рисуемую строку */
string = XGetDefault (display, "example", "MyResource"); . . . . .
. .
XDrawString ( display, window, gc, 10, 50, string,  strlen (string)
); . . . . .
```

# Доступ к ресурсам программ

После изменения файла ".Xdefaults" он должен быть обработан программой xrdp для того, чтобы X сервер включил в свою таблицу обновленные ресурсы.

Функция XGetDefault() проста в обращении, но не достаточно гибка. Так, например, с ее помощью нельзя прочитать содержимое произвольного файла ресурсов. Есть другие более развитые возможности.

Вызов функции XrmInitialize() инициализирует менеджер ресурсов. Обращение к этой функции предшествует вызовам остальных функций.

# Функция XrmParseCommand()

```
XrmParseCommand (XrmDatabase *db, XrmOptionRec *optRec,  
int optRecNum, char *programName, int argc, char **argv);
```

сканирует строку, с помощью которой вызвана программа, и "достаёт" из нее ресурсы и их значения, при этом создается специальная структура данных - база данных ресурсов. Ресурсы и их значения помещаются в нее. Указатель на базу данных передается программе через переменную **db**. Параметр **programName** содержит имя программы, **argc** - число опций в командной строке, **argv** - сами опции. Аргумент **optRec** определяет, как разбирать командную строку. **optRecNum** задает число элементов массива **optRec**.

# Пример программы

В примере, приводимом ниже, определяется, что в командной строке опция `"-bg"` задает цвет фона; `"-fg"` - цвет переднего плана, а опция `"-xrm"` позволяет задать в командной строке любой ресурс программы.

```
XrmOptionDescRec rOptRec[ ] = {
  { "-bg", "*background", XrmoptionSepArg, "Red" },
  { "-fg", "*foreground", XrmoptionSepArg, "White" },
  { "-xrm", NULL, XrmoptionResArg, NULL }, };
XrmDatabase rDB;
void main (int argc, char **argv) { .....
  XrmInitialize( );
  XrmParseCommand (&rDB, rOptRec,
                  sizeof (rOptRec) / sizeof (rOptRec[0]),
                  argv[0], argc, argv); ..... }
```

## Функция XrmGetFileDataBase()

Функция XrmGetFileDataBase() считывает ресурс с именем resName и классом resClass из базы данных \*db. После возврата resType есть указатель на строку, указывающую тип ресурса. На само значение ресурса указывает resVal.

Функция XrmPutResource() сохраняет ресурс в базе данных. XrmPutFileDatabase() записывает базу данных ресурсов в файл.