

## 8. X WINDOW SYSTEM – ГРАФИЧЕСКАЯ ОКОННАЯ СИСТЕМА

### 8.1 Общие сведения

Современные операционные системы не обходятся без подсистемы графического интерфейса пользователя (GUI). UNIX с самого начала позиционировалась как гибкая система, пусть даже в ущерб лёгкости использования. Тем не менее есть несколько причин использования GUI и в системах типа UNIX. Например, графический интерфейс облегчает управление традиционными для UNIX функциями многозадачности. Кроме того, некоторые виды информации удобнее представляются в графической форме, а некоторые виды только в графической форме и могут быть представлены. Исторически так сложилось, что различные элементы операционных систем типа UNIX разрабатывались в университетах (например, дистрибутивы BSD). В свою очередь X Window System (или просто «X») – основа графического интерфейса большинства операционных систем типа UNIX – также является университетской разработкой, а именно частью проекта Athena, разработанного в 1984 году в Массачусетском институте технологий (MIT).

UNIX изначально была многопользовательской сетевой операционной системой, позволяющей пользователям работать с ней с удалённого терминала. Подобные принципы легли в основу при создании GUI для UNIX. Фактически X имеет достаточно сложную структуру, что нередко упоминают как недостаток. Однако при всей своей сложности X является чрезвычайно гибкой системой, что позволило ей распространиться на громадном количестве платформ.

С 1997 года разработка X координируется The Open Group. Спецификация интерфейса X свободно доступна (последняя версия протокола, выпущенная 23.09.2008, обозначается X11R7.4). В GNU/Linux и в некоторых других ОС до 2004 г. использовалась модификация X под названием XFree86. В 2004 г. после выпуска очередной версии XFree86 (версия 4.4) под ограниченной лицензией, которую многие проекты, полагавшиеся на X, посчитали неприемлемой, представители X.Org и freedesktop.org основали фонд X.Org Foundation для разработки собственной реализации X. Именно эта реализация (X.Org Server) используется в настоящее время в большинстве дистрибутивов GNU/Linux.

Система X реализована по архитектуре «клиент–сервер». Графические приложения являются клиентами. Они соединяются с X-сервером, выполняют запросы и получают информацию от него. Клиентам надо лишь знать протокол общения с сервером, но совершенно не требуется информация о реальной аппаратной реализации графической подсистемы. Клиенты выполняют запросы типа «нарисуй линию с указанными координатами», «выведи такой-то текст определённым шрифтом в указанной позиции» и т. п. Такая архитектура позволяет разнести клиентов и сервер на разные машины. Например, можно запустить какую-нибудь вычислительную задачу на ЭВМ типа Cray, систему управления базами данных – на сервере под управлением Solaris, почтовую программу – на небольшом сервере BSD, а программу графического моделирования – на SGI-сервере, причём все эти программы могут выводить результаты своей работы у вас дома на рабочей станции под управлением GNU/Linux.

Следовательно, X-сервер обслуживает реальный графический дисплей. X-сервер запускается на том компьютере, за которым работает оператор. Именно обязанностью X-сервера является фактическое взаимодействие с пользователем, получение событий от мыши и клавиатуры. Однако, хотя X-сервер обеспечивает механизмы манипулирования областью отображения клиента (окном), но он не предоставляет средств для этого. Где отобразить окно клиента, средства изменения окна клиента, средства оформления окна клиента (заголовок, рамка и т. п.) – все эти задачи решаются специальной программой – «диспетчером окон». Диспетчер окон – это специальный X-клиент, не являющийся по своей сути частью X Window System, но наделённый определёнными привилегиями. Существует огромное количество диспетчеров окон для X: twm, fvwm, icewm, afterstep, enlightenment и sawfish (используются в GNOME), kwm (используется в KDE) и многие другие.

### 8.2 Запуск X-сервера и графических приложений

X может стартовать при запуске операционной системы. В таком случае для регистрации пользователя в системе используется программа **xdm** или аналогичная ей. Если X не стартует автоматически, то графическую подсистему можно запустить с помощью специального сценария **startx**:

```
startx [[клиент] опции] [-- [сервер] опции]
```

Этот сценарий выполняет команду **xinit**, запуская указанный X-сервер и соединяя с ним первоначального клиента. Кроме того, в этом сценарии выполняются различные настройки X-сервера. Если клиент или сервер не указаны, то используются настройки из файлов конфигурации `~/.xinitrc` и `~/.xserverrc`. Обычно эта команда вызывается без параметров, что обеспечивает запуск подходящего X-сервера и стандартного диспетчера окон в качестве начального клиента на дисплее под номером 0. Как правило, X-сервер запускается на свободном виртуальном терминале системной консоли. Переключение между виртуальными терминалами в графическом режиме осуществляется при помощи комбинаций [Ctrl]+[Alt]+[Fn] (обычно  $n=1..6$  – для текстовых

терминалов и  $n=7, \dots$  – для графических; комбинации [Alt]+[Fn] зарезервированы для использования X-клиентами).

Графическим программам (X-клиентам) необходимо знать адрес X-сервера, который задаётся в виде:

*hostname:displayname.screenname*

- *hostname* – имя компьютера, на котором запущен X-сервер (в случае локального запуска может быть пустым);
- *displayname* – номер дисплея (под «дисплеем» понимается группа мониторов, разделяющая одну клавиатуру и мышь);
- *screenname* – номер экрана (номер монитора в группе).

Типичное значение адреса X-сервера – «:0.0». Адрес X-сервера обычно передаётся X-клиенту диспетчером окон через среду окружения в переменной DISPLAY. Другой способ передачи адреса – в параметре командной строки *-display*. Если переменная окружения DISPLAY не определена и параметр *-display* не задан, то X-клиент откажется запускаться. Примеры:

```
DISPLAY=":0.0" xcalc
xclock -display huba.buba.com:10.0
```

Допускается разделение несколькими X-серверами одного физического дисплея, т. е. запуск нескольких серверов на одном компьютере, но при этом им надо назначать различные номера логических дисплеев. Это достигается при помощи опции *:N* ( $N=0,1,\dots$ ) при старте сервера:

```
startx -- :1
```

При этом каждый логический дисплей занимает отдельный виртуальный терминал системной консоли. Для логических дисплеев используется своя нумерация, не совпадающая с нумерацией виртуальных терминалов. Так, в этом примере X-серверу присваивается логический дисплей 1, который занимает первый свободный виртуальный терминал (например, № 7).

Среди прочих параметров сервера пользователь может также указать с помощью параметра *-depth* глубину цвета (количество бит на пиксел) в графическом режиме:

```
startx -- -depth 8
startx -- :2 -depth 32
```

Разрешение экрана можно поменять в процессе работы при помощи комбинаций<sup>10</sup> [Ctrl]+[Alt]+[+] и [Ctrl]+[Alt]+[-].

Во многих диспетчерах окон комбинация [Ctrl]+[Alt]+[Esc] запускает программу **xkill**. Эта программа по щелчку мыши на окне того или иного приложения определяет процесс, которому принадлежит это окно, и вынуждает сервер разорвать соединение с этим клиентом, обеспечивая таким образом принудительное завершение работы того или иного приложения.

Другая полезная комбинация – [Ctrl]+[Alt]+[BackSpace]. Эта комбинация вынуждает X-сервер закрыть соединения со всеми клиентами и завершиться. Будьте очень осторожны: все несохраненные данные в программах-клиентах будут потеряны! Администратор системы может сконфигурировать X-сервер так, чтобы эта комбинация игнорировалась.

### 8.3 Ресурсы графических приложений

Все клиенты, использующие стандартную библиотеку Xlib, поддерживают параметры командной строки, влияющие на отображение их окна. Кроме уже упомянутого параметра *-display* распознаются следующие параметры:

- *-title имя* – задать заголовок окна;

```
xlogo -title "X rulezzz 4ever!!!"
```

- *-iconic* – запускать свёрнутым;

```
xload -iconic
```

- *-geometry [ширинаxвысота][±xpos±ypos]* – позиция и размеры окна программы (“+” соответствует отсчётам от левой или верхней грани экрана, а “-” от правой или нижней), единица измерения (символы или пиксели) зависит от приложения;

```
xlogo -geometry 100x200+100+200
xcalc -geometry -50+50
xterm -geometry 80x43
```

- *-background цвет* или *-bg цвет* – установка цвета фона;
- *-foreground цвет* или *-fg цвет* – установка основного цвета;

<sup>10</sup> Клавиши [+] и [-] на дополнительной клавиатуре («серые»).

```
xcalc -background orange -fg blue
xclock -bg rgb:0/FF/0 -foreground rgb:FF/80/80
xfontsel -background '#0080FF' -fg '#FF0000'
```

- *-font шрифт* или *-fn шрифт* – установка основного шрифта (как описывается тот или иной шрифт можно определить при помощи программы **xfontsel**)

```
xclock -digital -font '-cronyx-helvetica-medium-r-*-*-koi8-r'
xclock -digital -font '-cronyx-courier-bold-o-*-*-koi8-r'
xcalc -font '-adobe-times-*-*-koi8-r'
```

Настройки такого рода называются *описанием ресурсов* приложения. Значения по умолчанию хранятся в базе данных ресурсов X-сервера. При запуске X-сервера из сценария **startx** эта база данных заполняется из общесистемных файлов описания ресурсов (/usr/lib/X11/app-defaults/\*) и пользовательских файлов настроек (~/.Xdefaults или ~/.Xresources). Эта база данных может изменяться интерактивно при помощи команды **xrdb**. Например, чтобы добавить в базу данных ресурсов описания из файла appres, надо выполнить команду:

```
xrdb -merge appres
```

Если имя файла не указано, описания ресурсов читаются со стандартного ввода. Просмотреть текущую базу ресурсов можно, задав ключ *-query*:

```
$ xrdb -query
Emacs*menubar*background: #DCDCDC
XTerm*font: -adobe-courier-*-*-iso8859-1
XTerm*background: #FFFFFF
XTerm*foreground: #000000
$ xrdb -merge
XTerm*font: -misc-fixed-*-*-koi8-r
```

В обозначении атрибутов графических элементов приложения строчные и прописные буквы различаются! (Например, для **xterm** должно быть «XTerm...», а для **xcalc** – «xcalc...»).

## 8.4 Контроль доступа (авторизация клиентов)

X-сервер может контролировать доступ клиентов при помощи одного из пяти механизмов.

**Host Access Control** – простой контроль по имени хоста. X-сервер хранит список хостов, доступ клиентам с которых безусловно разрешается (причём владелец процесса X-клиента не проверяется). Такая система авторизации применима в том случае, если все пользователи доверяют друг другу. Изменение списка хостов осуществляется при помощи команды **xhost**. При использовании более безопасных средств контроля доступа список хостов должен быть пуст, тогда доступ к X-серверу будут получать только авторизованные клиенты.

**MIT-MAGIC-COOKIE-1** – при использовании этого метода авторизации клиент обязан послать X-серверу 128-битный секретный код (*magic cookie*). Если код, посланный клиентом, совпадает с тем, который использует сервер, доступ клиенту разрешается. Пользовательская копия кода доступа обычно хранится в бинарном файле ~/.Xauthority. Операции с этим файлом производятся при помощи команды **xauth**. Программа **xdm** автоматически сохраняет *magic cookie* в каталоге пользователя и передаёт X-серверу. При локальном запуске X-сервера при помощи сценария **startx** генерируется новый *magic cookie*, помещается в пользовательскую базу кодов, а X-серверу для обеспечения авторизации в опции *-auth* передаётся пользовательский файл с кодом. *Magic cookie* посылается по сети в открытом виде, поэтому такую схему авторизации не следует применять в сетях, где вероятен перехват сетевых пакетов.

**XDM-AUTHORIZATION-1** – этот механизм аналогичен MIT-MAGIC-COOKIE-1, за исключением того, что при передаче секретного кода используется шифрование при помощи симметричного алгоритма DES.

**SUN-DES-1** – авторизация на основе безопасных вызовов удалённых процедур (secure RPC). При помощи механизма secure RPC обеспечивается аутентификация пользователя. Контроль доступа осуществляется по регистрационным именам пользователей. Этот механизм применим только в системах, поддерживающих Secure RPC.

**MIT-KERBEROS-5** – авторизация по протоколу Kerberos 5.

Как правило, удалённые X-сервера используются только в пределах локальных вычислительных сетей, поэтому именно первые два механизма авторизации широко распространены, хотя они и менее безопасны, чем три остальных.

Если X-клиент и X-сервер работают на разных машинах, то, очевидно, наиболее полезна ситуация, когда удалённый X-клиент использует X-сервер на вашей машине (а не наоборот). Именно в этой ситуации обычно используют XDMCP (X Display Manager Control Protocol) и SSH X11 Forwarding.

Когда локальный X-сервер настраивается на соединение с удалённой машиной по протоколу XDMCP, удалённая машина выводит на вашем X-сервере окно регистрации. После успешной регистрации вы можете работать с удалённой машиной, причём все запущенные вами на удалённой машине X-клиенты будут соединяться с вашим X-сервером. В качестве XDMCP-сервера на удалённой машине обычно выступает программа **xdm**

или аналогичная ей (**gdm, kdm**). Для запуска X-сервера в режиме XDMCP-клиента ему надо передать ключ *-query* с адресом XDMCP-сервера:

X -query <i>huba.buba.com</i>
-------------------------------

Авторизацию обеспечивает механизм MIT-MAGIC-COOKIE-1. X-сервер будет использовать протокол XDMCP для согласования секретного кода. Сам код будет помещён XDMCP-сервером в вашу базу кодов на удалённой машине (~/.Xauthority).

Протокол XDMCP уязвим к перехвату сетевых пакетов, т. к. информация передаётся в открытом виде. Если требуется более безопасный механизм, следует использовать SSH X11 Forwarding.

На своём X-сервере вы запускаете SSH-клиент и регистрируетесь на удалённой машине. Если механизм X11 Forwarding поддерживается обеими сторонами, то на удалённой машине SSH создаёт виртуальный «прокси» X-сервер и устанавливает переменную окружения DISPLAY так, чтобы удалённые X-клиенты соединялись с этим виртуальным «прокси» X-сервером. На самом деле, данные, поступающие для этого виртуального сервера, передаются SSH по безопасному каналу вашему локальному X-серверу. При этом SSH генерирует для виртуального сервера свой *magic cookie*, а при перенаправлении данных, заменяет его на реальный секретный код для вашего сервера, так что *magic cookie* вашего X-сервера не хранится на удалённой машине и в открытом виде по сети не передаётся.

### Контрольные вопросы и задания

1. При помощи команды **startx** запустите несколько X-серверов. Определите, какие виртуальные терминалы ими заняты. Сколько X-серверов вам удалось запустить?
2. Определите, какие адреса соответствуют X-серверам на разных консолях. Для этого с помощью средств вашего диспетчера окон запустите какой-нибудь эмулятор терминала для X (**xterm, rxvt, konsole** и т. п.). В окне эмулятора терминала выведите значение переменной окружения DISPLAY.
3. Из первого виртуального терминала (tty1) запустите на разных X-серверах программы **xcalc** и **xclock**.
4. Поэкспериментируйте с различными параметрами командной строки, описывающими вид окна, для программ **xcalc, xlogo, xclock, xfontsel, xload, xterm, rxvt**.
5. Измените базу данных ресурсов так, чтобы программа **xcalc** по умолчанию запускалась с голубым фоновым цветом, белым основным цветом и использовался шрифт *misc-fixed* с кодировкой *iso8859-1*.
6. Измените пользовательский файл описания ресурсов так, чтобы программа **xterm** запускалась с зелёным фоновым цветом, с красным основным цветом, шрифт – *stoux-courier* с кодировкой *koï8-r*, размер окна по умолчанию – 100x43 симв. Перезапустите X-сервер и убедитесь, что новые описания ресурсов вступили в силу.
7. Попросите своего соседа, чтобы он для своего X-сервера при помощи команды **xhost** разрешил доступ с вашего компьютера. Запустите программу **xclock**, передав адрес X-сервера вашего соседа в переменной DISPLAY, и программу **xcalc**, передав адрес X-сервера вашего соседа в аргументе командной строки *-display*. Попросите соседа отключить доступ с вашего компьютера и сравните результат.
8. Выясните у своего соседа 128-битный код доступа (*magic cookie*) для его X-сервера. Добавьте этот код в свою базу *cookies* при помощи команды **xauth**. Запустите программы **xlogo** и **xload**, указав им адрес сервера вашего соседа.
9. Зайдите на указанную преподавателем машину при помощи **telnet** или **ssh** (X11 Forwarding запрещён). Запустите на удалённой машине несколько X-клиентов так, чтобы они выводили информацию на ваш X-сервер (настройте доступ для вашего X-сервера сначала при помощи Host Access Control, а затем при помощи MIT-MAGIC-COOKIE-1).
10. Запустите ваш X-сервер в режиме XDMCP-клиента. В этом режиме запустите на XDMCP-сервере несколько X-клиентов.
11. Ознакомьтесь с работой SSH X11 Forwarding.