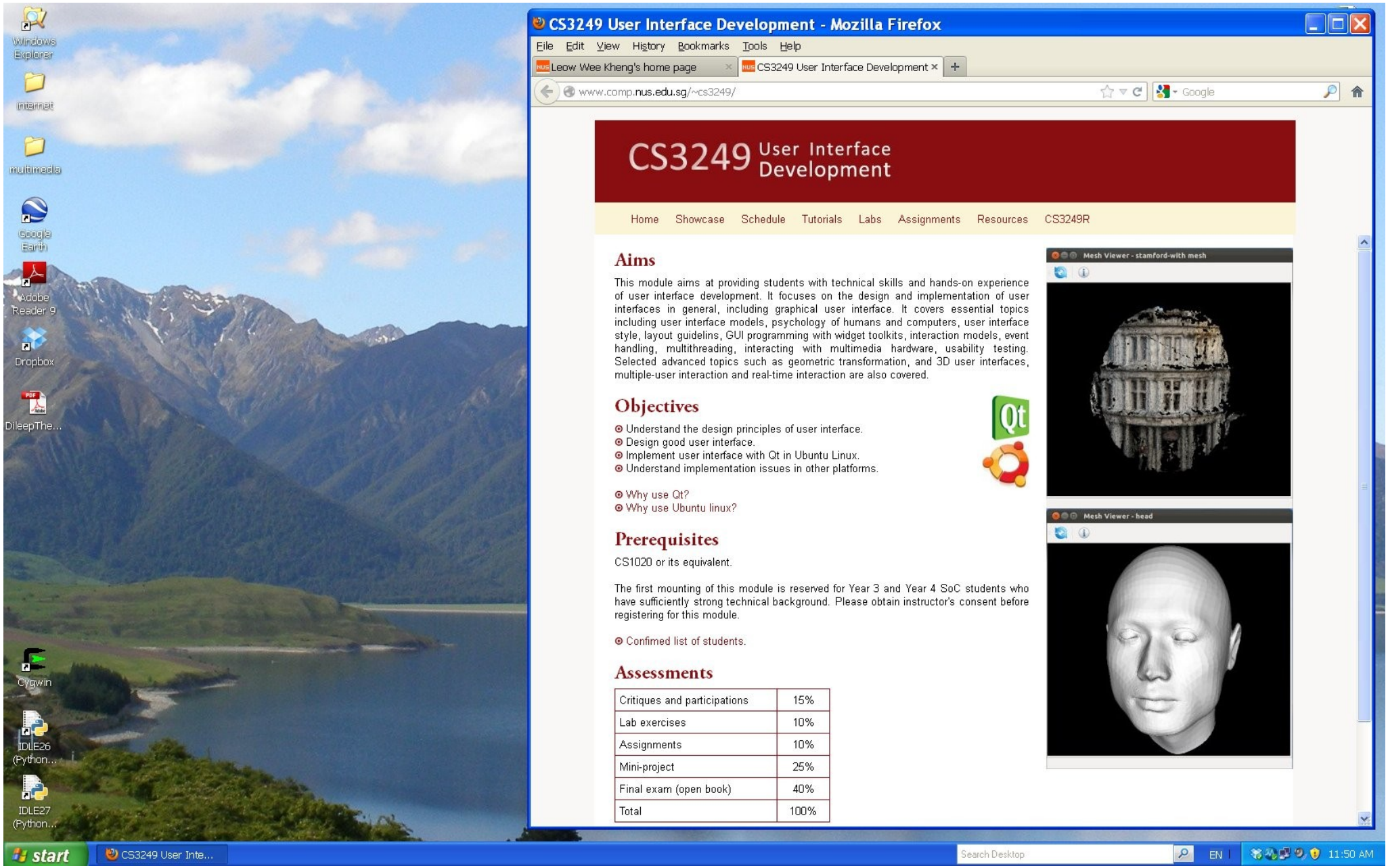
Leow Wee Kheng
CS3249 User Interface Development

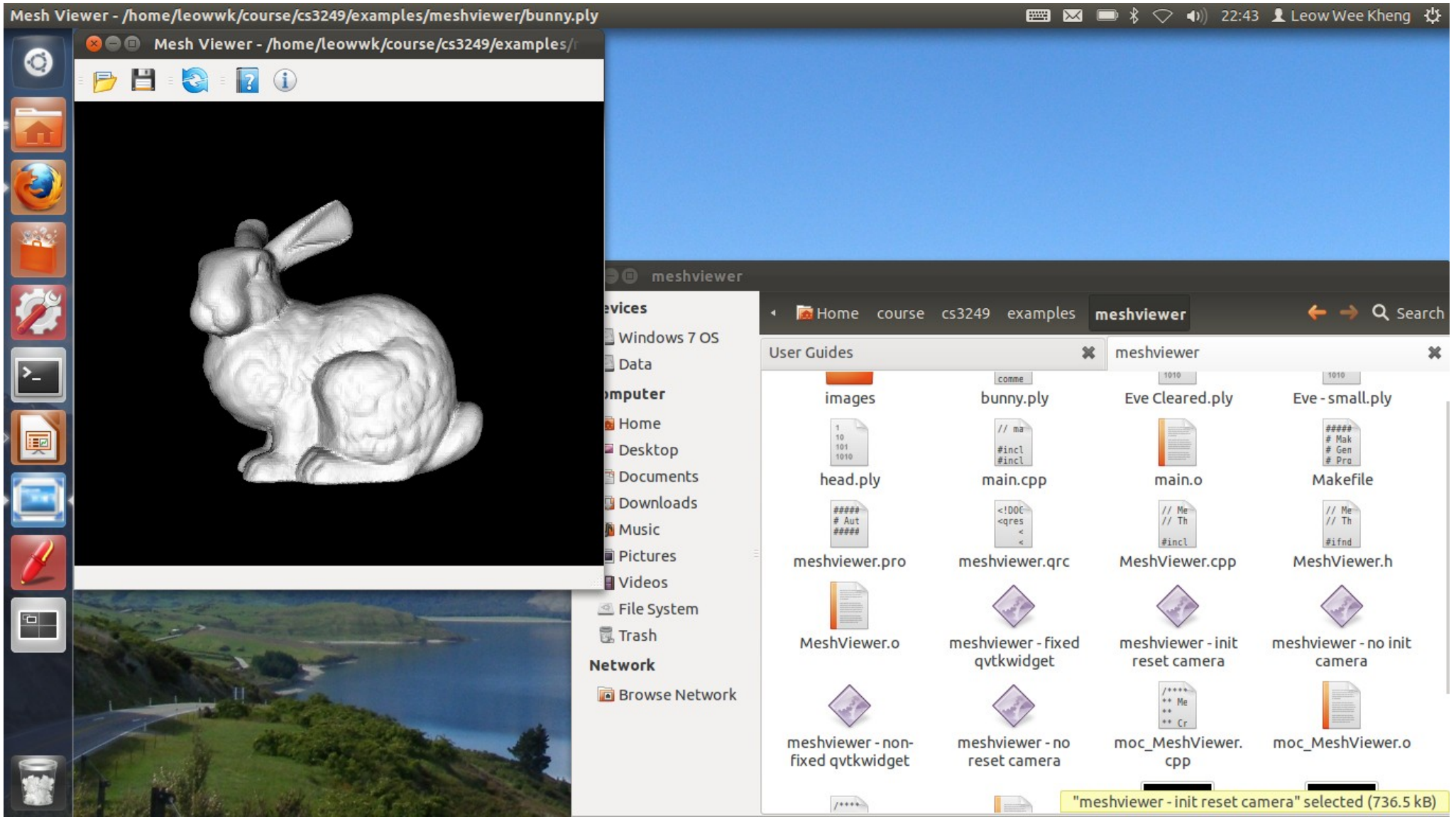
Windowing Systems

Modern computers come with graphical user interfaces...

Windows XP

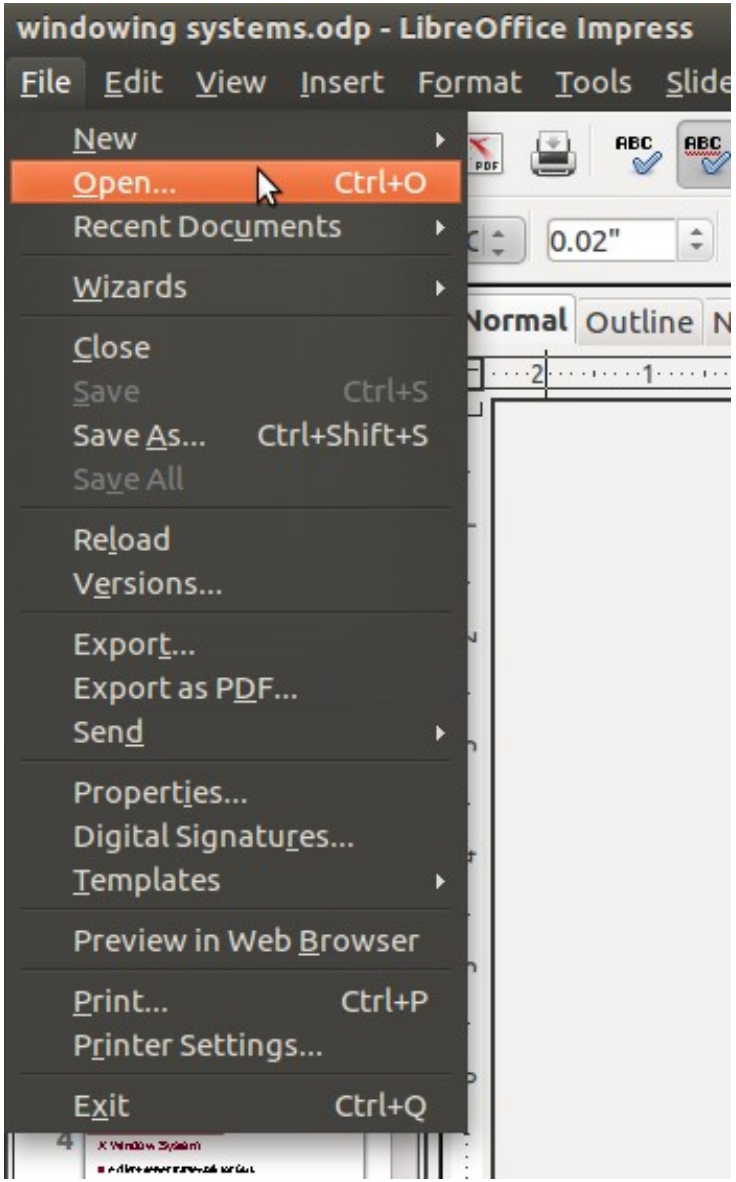

CS3249 User Interface Development - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Leow Wee Kheng's home page
CS3249 User Interface Development
www.comp.nus.edu.sg/~cs3249/
CS3249 User Interface Development
Home Showcase Schedule Tutorials Labs Assignments Resources CS3249R
Aims
This module aims at providing students with technical skills and hands-on experience of user interface development. It focuses on the design and implementation of user interfaces in general, including graphical user interface. It covers essential topics including user interface models, psychology of humans and computers, user interface style, layout guidelins, GUI programming with widget toolkits, interaction models, event handling, multithreading, interacting with multimedia hardware, usability testing. Selected advanced topics such as geometric transformation, and 3D user interfaces, multiple-user interaction and real-time interaction are also covered.
Objectives
Understand the design principles of user interface.
Design good user interface.
Implement user interface with Qt in Ubuntu Linux.
Understand implementation issues in other platforms.
Why use Qt?
Why use Ubuntu linux?
Prerequisites
CS1020 or its equivalent.
The first mounting of this module is reserved for Year 3 and Year 4 SoC students who have sufficiently strong technical background. Please obtain instructor's consent before registering for this module.
Confimed list of students.
Assessments
Critiques and participations 15%
Lab exercises 10%
Assignments 10%
Mini-project 25%
Final exam (open book) 40%
Total 100%
Mesh Viewer - stamford-with mesh
Mesh Viewer - head
start
CS3249 User Inte...
Search Desktop
EN
11:50 AM

Ubuntu


Mesh Viewer - /home/leowwk/course/cs3249/examples/meshviewer/bunny.ply
22:43
Leow Wee Kheng
Mesh Viewer - /home/leowwk/course/cs3249/examples/
meshviewer
Home
course
cs3249
examples
meshviewer
Search
Windows 7 OS
Data
Home
Desktop
Documents
Downloads
Music
Pictures
Videos
File System
Trash
Network
Browse Network
User Guides
meshviewer
images
bunny.ply
Eve Cleared.ply
Eve - small.ply
head.ply
main.cpp
main.o
Makefile
meshviewer.pro
meshviewer.qrc
MeshViewer.cpp
MeshViewer.h
MeshViewer.o
meshviewer - fixed qvtkwidget
meshviewer - init reset camera
meshviewer - no init camera
meshviewer - non-fixed qvtkwidget
meshviewer - no reset camera
moc_MeshViewer.cpp
moc_MeshViewer.o
"meshviewer - init reset camera" selected (736.5 kB)

What is needed for GUI to work?

- What is needed for drop-down menu to work?
  - How to paint the menu?
  - Where is the mouse?
  - How to move the cursor?
  - Which part of the menu is the mouse pointing at?
  - Which mouse button is clicked?

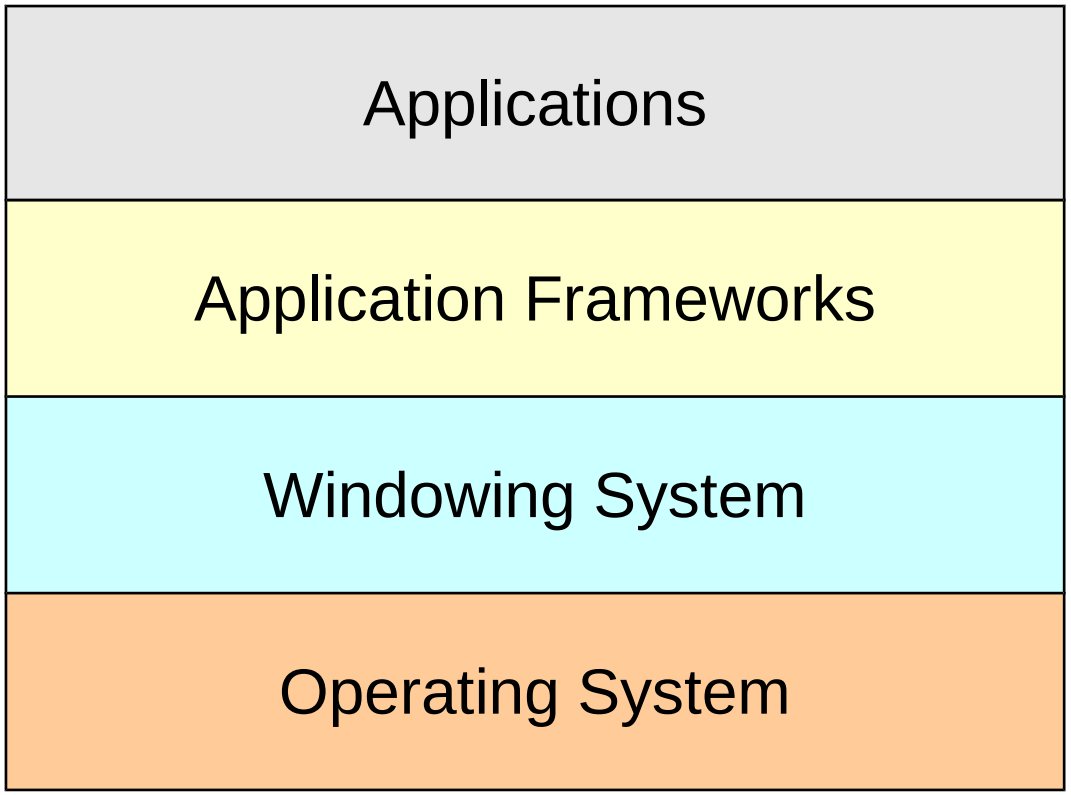


Supports needed:

- Capture user actions.
  - Low-level actions are captured by operating system.
  - Translated to higher-level actions by windowing system.
- Pass user actions to application codes.
  - Connect UI elements to application codes.

Windowing System

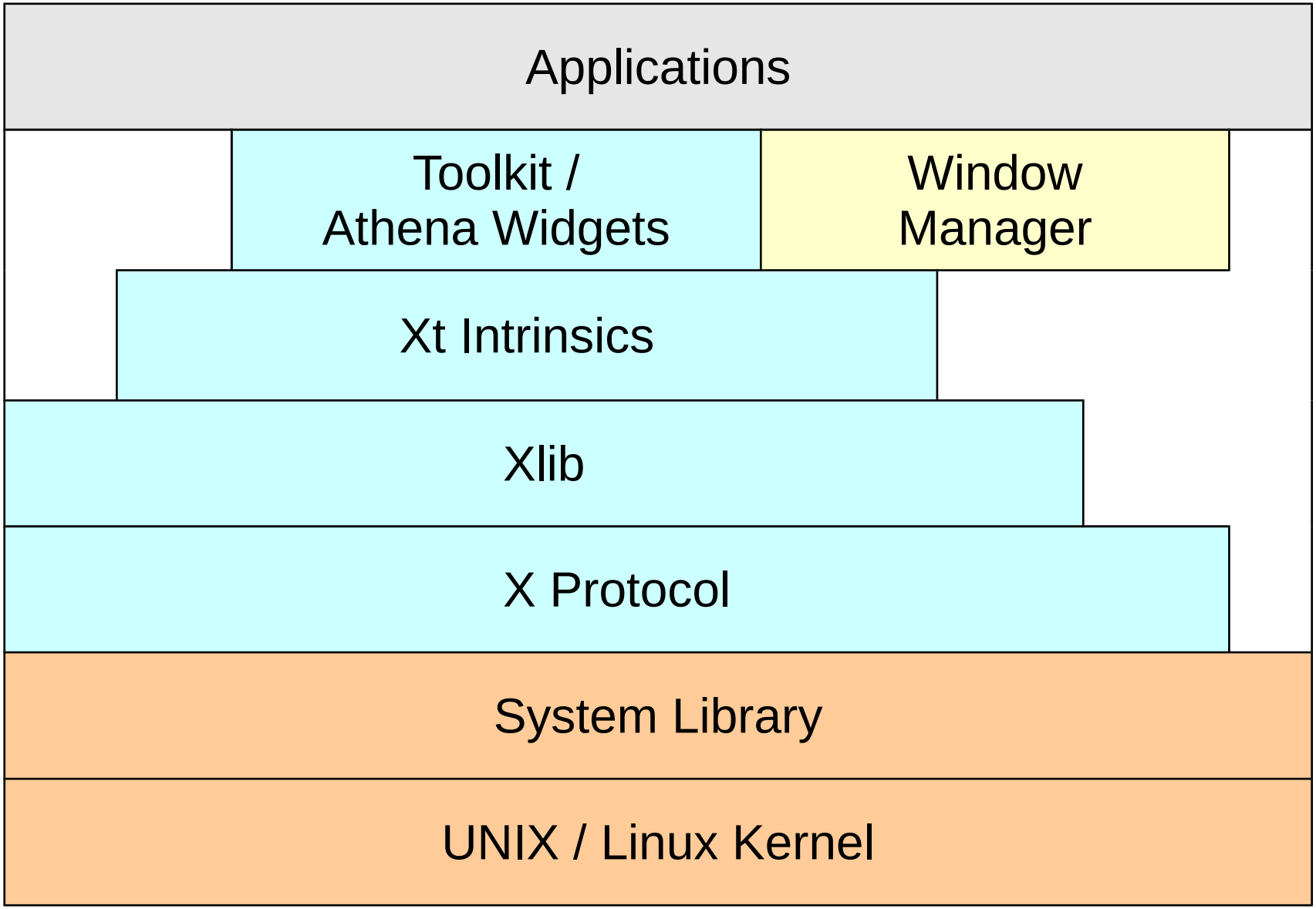
- Modern GUI works on top of a windowing system.
- Windowing system provides supports for
  - input & output devices thru operating system
  - high-level abstractions of graphical primitives
  - window management



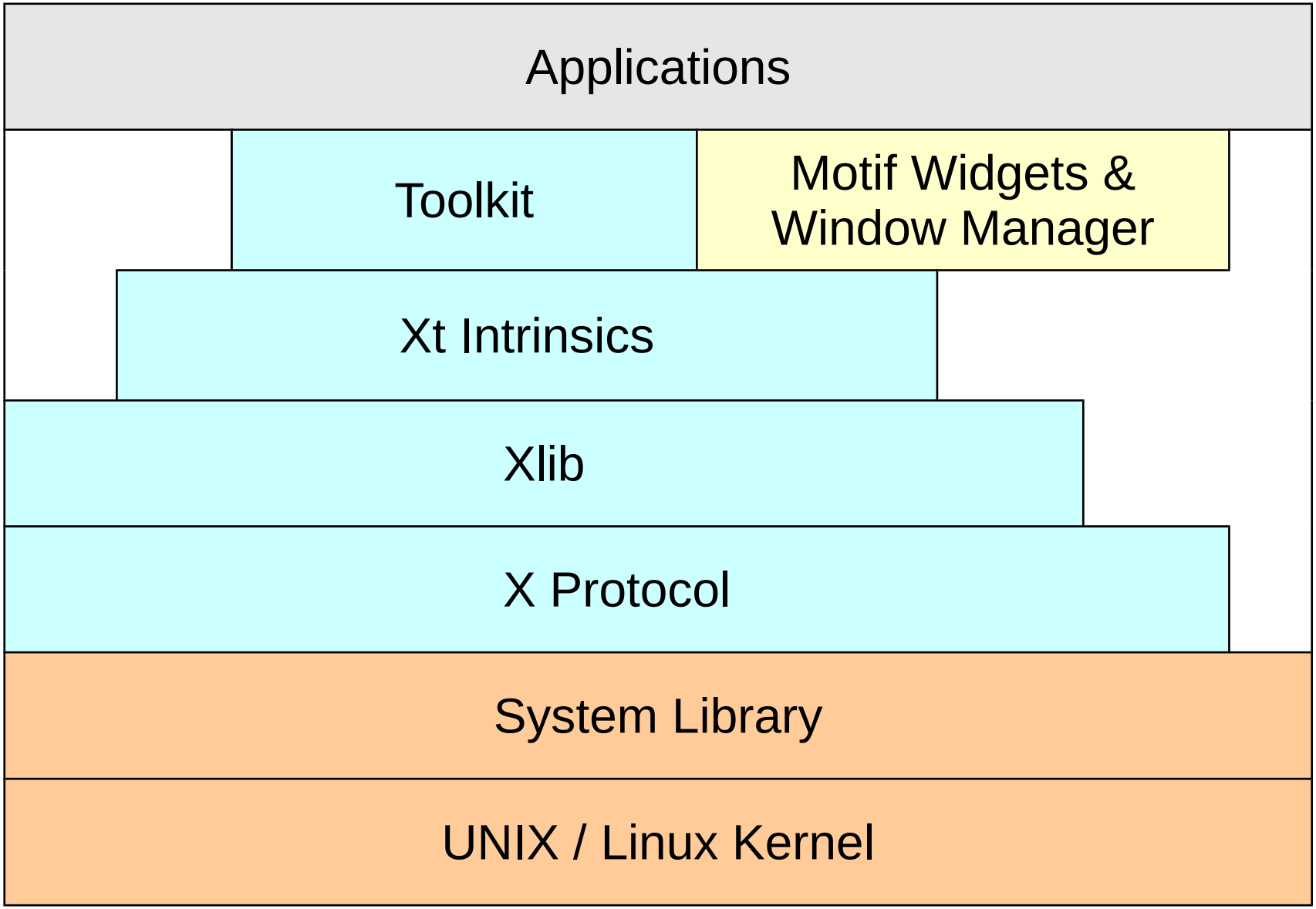
X Window System

- A client-server framework for GUI.
- X server
  - provide multi-client access over network
  - receive and understand client messages
  - send user responses to clients
  - perform all drawing
  - maintain data structures for resources
- Client applications can run on other hosts in network.

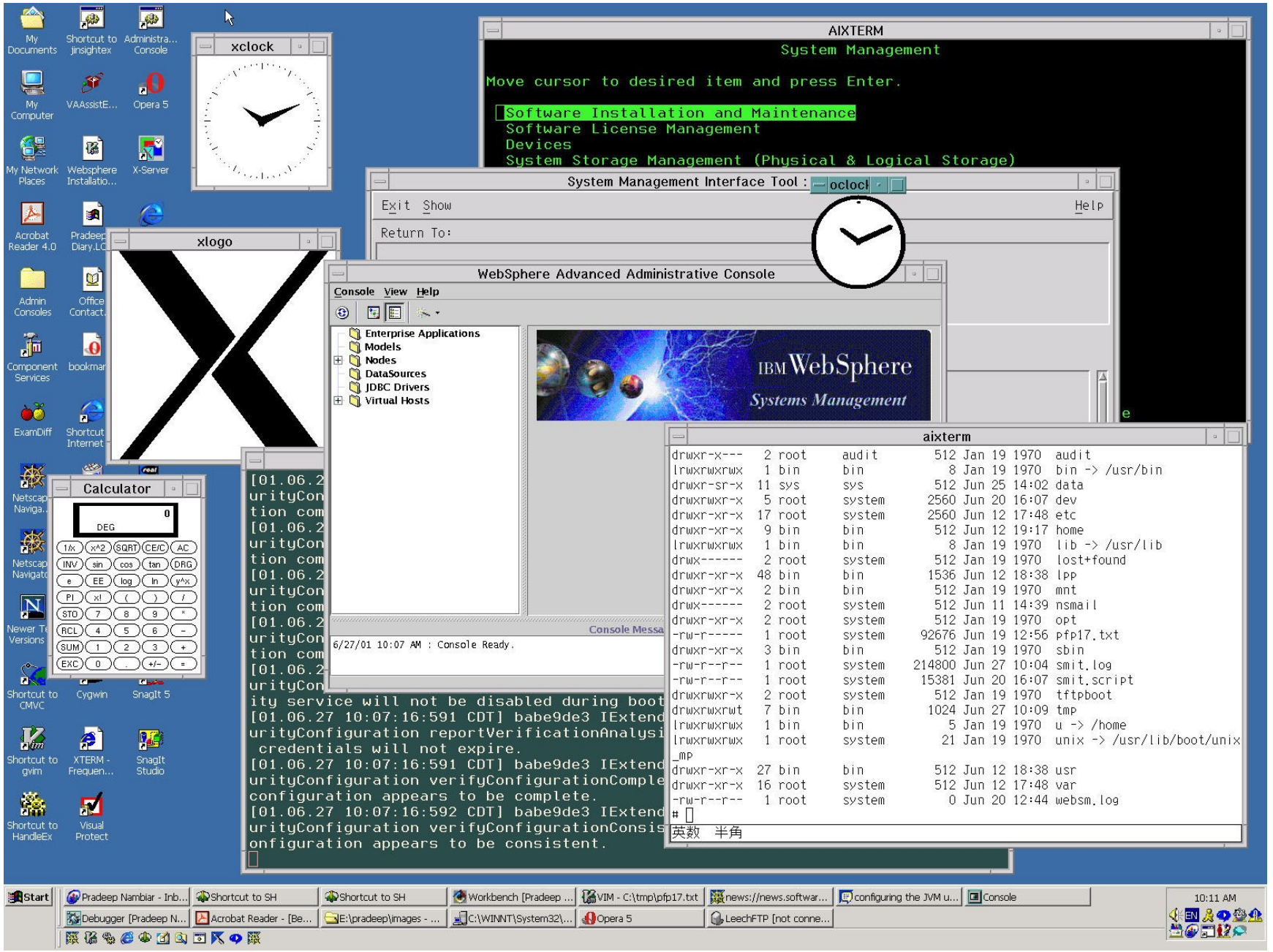
X System


Applications
Toolkit /
Athena Widgets
Window
Manager
Xt Intrinsics
Xlib
X Protocol
System Library
UNIX / Linux Kernel

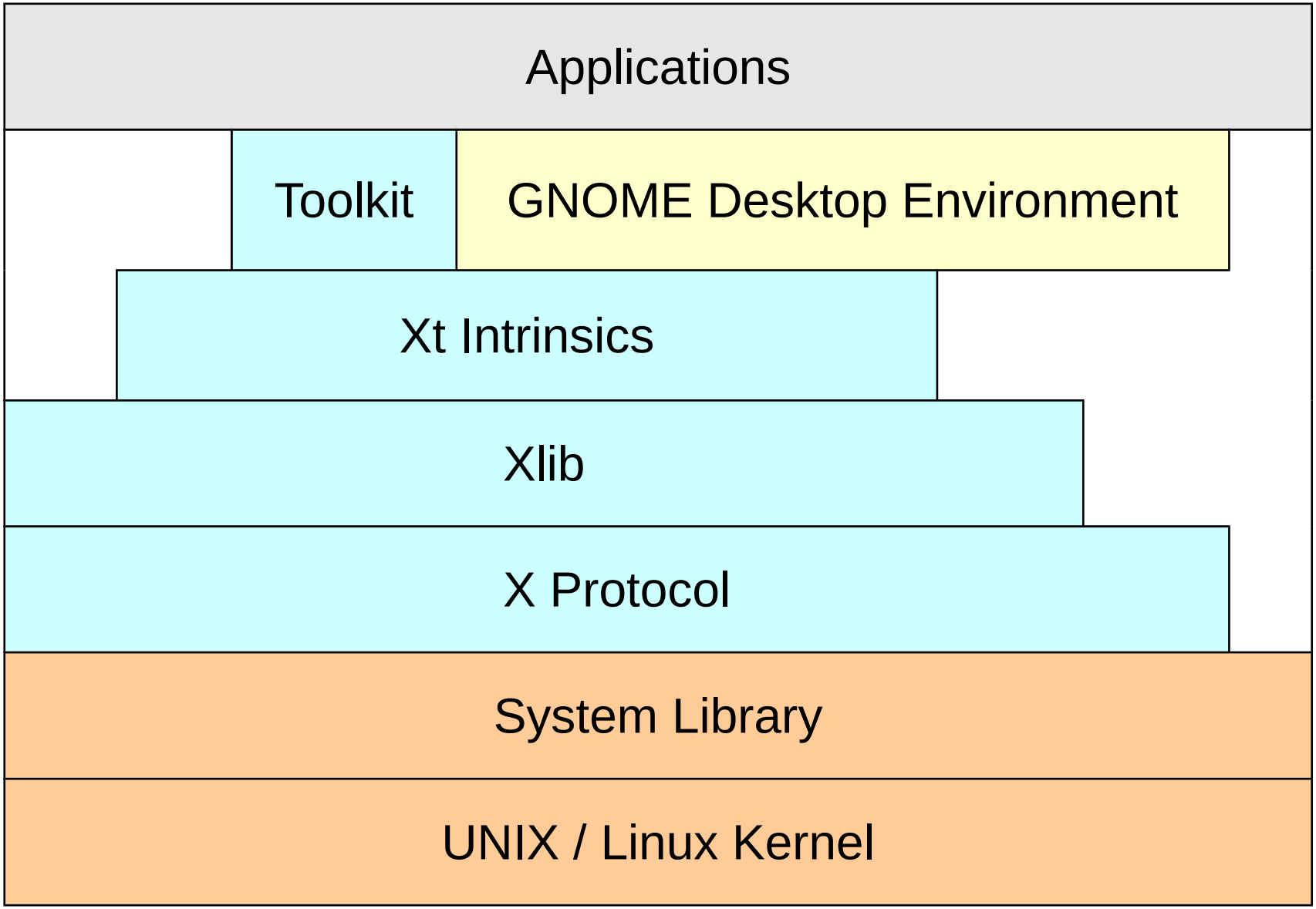
X / Motif


Applications
Toolkit
Motif Widgets &
Window Manager
Xt Intrinsics
Xlib
X Protocol
System Library
UNIX / Linux Kernel

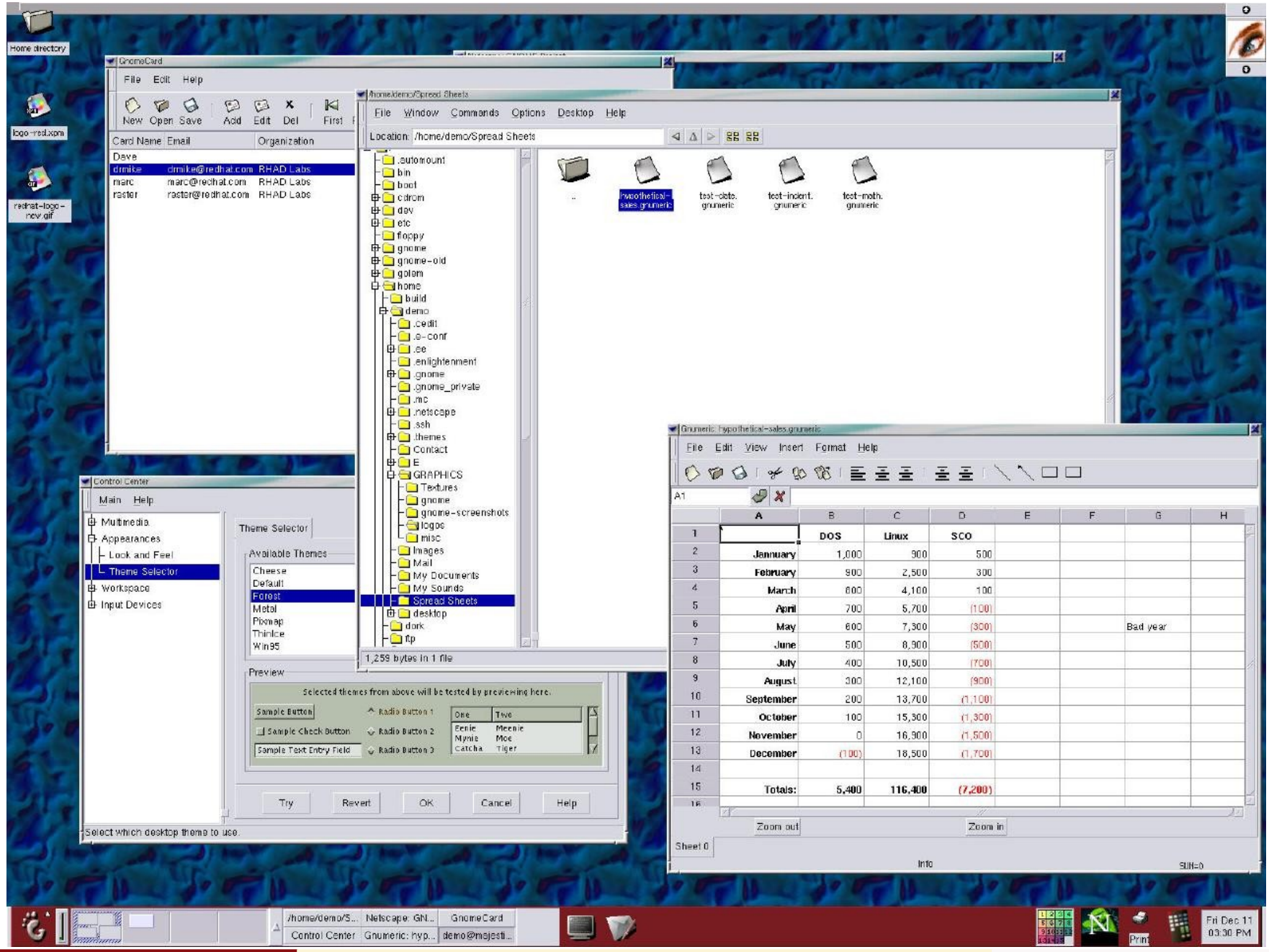
X with Motif


xclock
xlogo
AIXTERM
System Management
Move cursor to desired item and press Enter.
Software Installation and Maintenance
Software License Management
Devices
System Storage Management (Physical & Logical Storage)
System Management Interface Tool : oclock
Exit Show
Help
Return To:
WebSphere Advanced Administrative Console
Console View Help
Enterprise Applications
Models
Nodes
DataSources
JDBC Drivers
Virtual Hosts
IBM WebSphere
Systems Management
Console Messa
6/27/01 10:07 AM : Console Ready.
Calculator
aixterm
drwxr-x--- 2 root audit 512 Jan 19 1970 audit
lrwxrwxrwx 1 bin bin 8 Jan 19 1970 bin -> /usr/bin
drwxr-sr-x 11 sys sys 512 Jun 25 14:02 data
drwxrwxr-x 5 root system 2560 Jun 20 16:07 dev
drwxr-xr-x 17 root system 2560 Jun 12 17:48 etc
drwxr-xr-x 9 bin bin 512 Jun 12 19:17 home
lrwxrwxrwx 1 bin bin 8 Jan 19 1970 lib -> /usr/lib
drwx------ 2 root system 512 Jan 19 1970 lost+found
drwxr-xr-x 48 bin bin 1536 Jun 12 18:38 lpp
drwxr-xr-x 2 bin bin 512 Jan 19 1970 mnt
drwx------ 2 root system 512 Jun 11 14:39 nsmail
drwxr-xr-x 2 root system 512 Jan 19 1970 opt
-rw-r----- 1 root system 92676 Jun 19 12:56 pfp17.txt
drwxr-xr-x 3 bin bin 512 Jan 19 1970 sbin
-rw-r--r-- 1 root system 214800 Jun 27 10:04 smit.log
-rw-r--r-- 1 root system 15381 Jun 20 16:07 smit.script
drwxrwxr-x 2 root system 512 Jan 19 1970 tftpboot
drwxrwxrwt 7 bin bin 1024 Jun 27 10:09 tmp
lrwxrwxrwx 1 bin bin 5 Jan 19 1970 u -> /home
lrwxrwxrwx 1 root system 21 Jan 19 1970 unix -> /usr/lib/boot/unix_MP
drwxr-xr-x 27 bin bin 512 Jun 12 18:38 usr
drwxr-xr-x 16 root system 512 Jun 12 17:48 var
-rw-r--r-- 1 root system 0 Jun 20 12:44 websm.log
英数 半角
[01.06.27 10:07:16:591 CDT] babe9de3 IExtend
urityConfiguration reportVerificationAnalysi
credentials will not expire.
[01.06.27 10:07:16:591 CDT] babe9de3 IExtend
urityConfiguration verifyConfigurationComple
configuration appears to be complete.
[01.06.27 10:07:16:592 CDT] babe9de3 IExtend
urityConfiguration verifyConfigurationConsis
onfiguration appears to be consistent.
ity service will not be disabled during boot
Start
10:11 AM

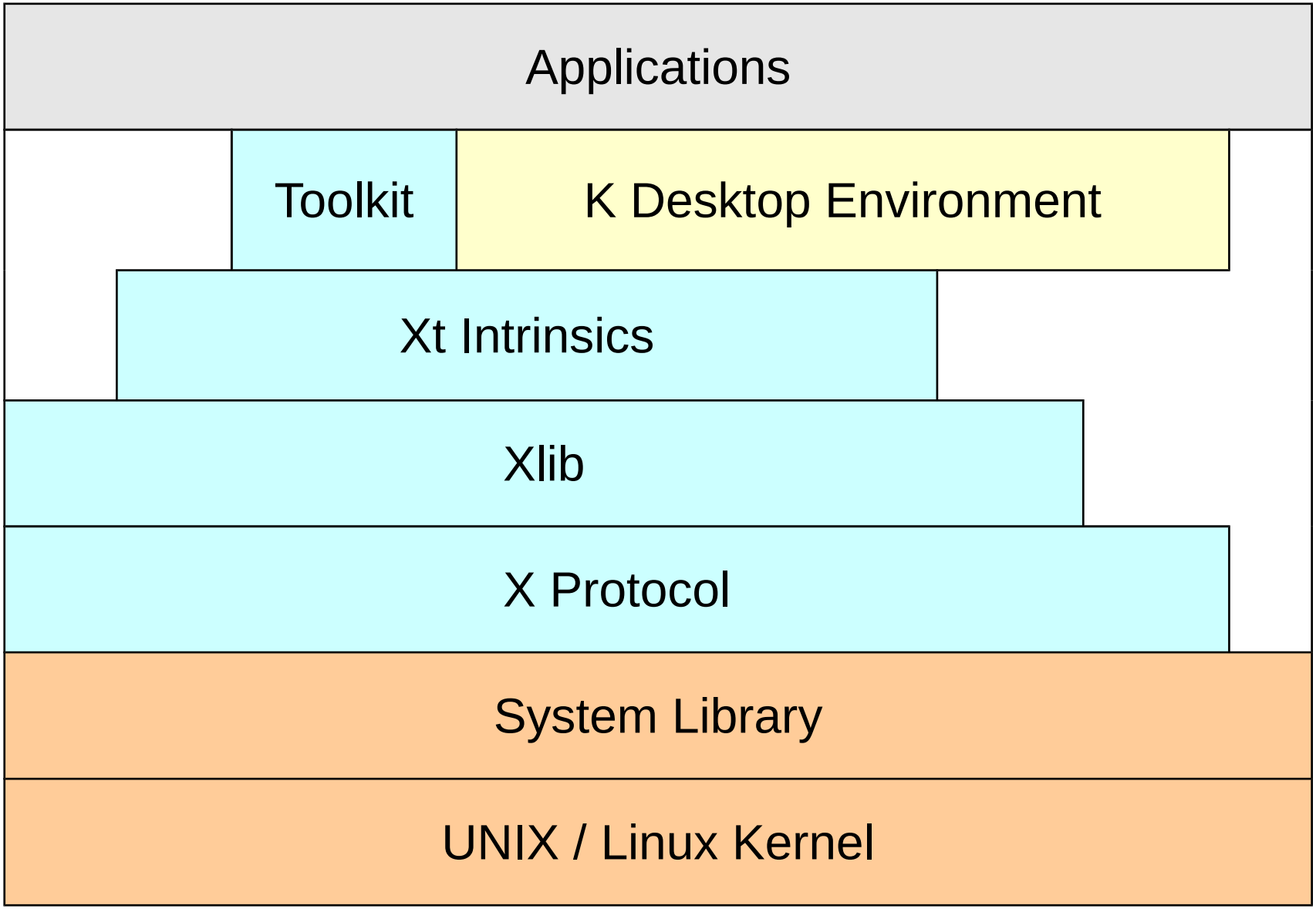
X with GNOME


Applications
Toolkit
GNOME Desktop Environment
Xt Intrinsics
Xlib
X Protocol
System Library
UNIX / Linux Kernel

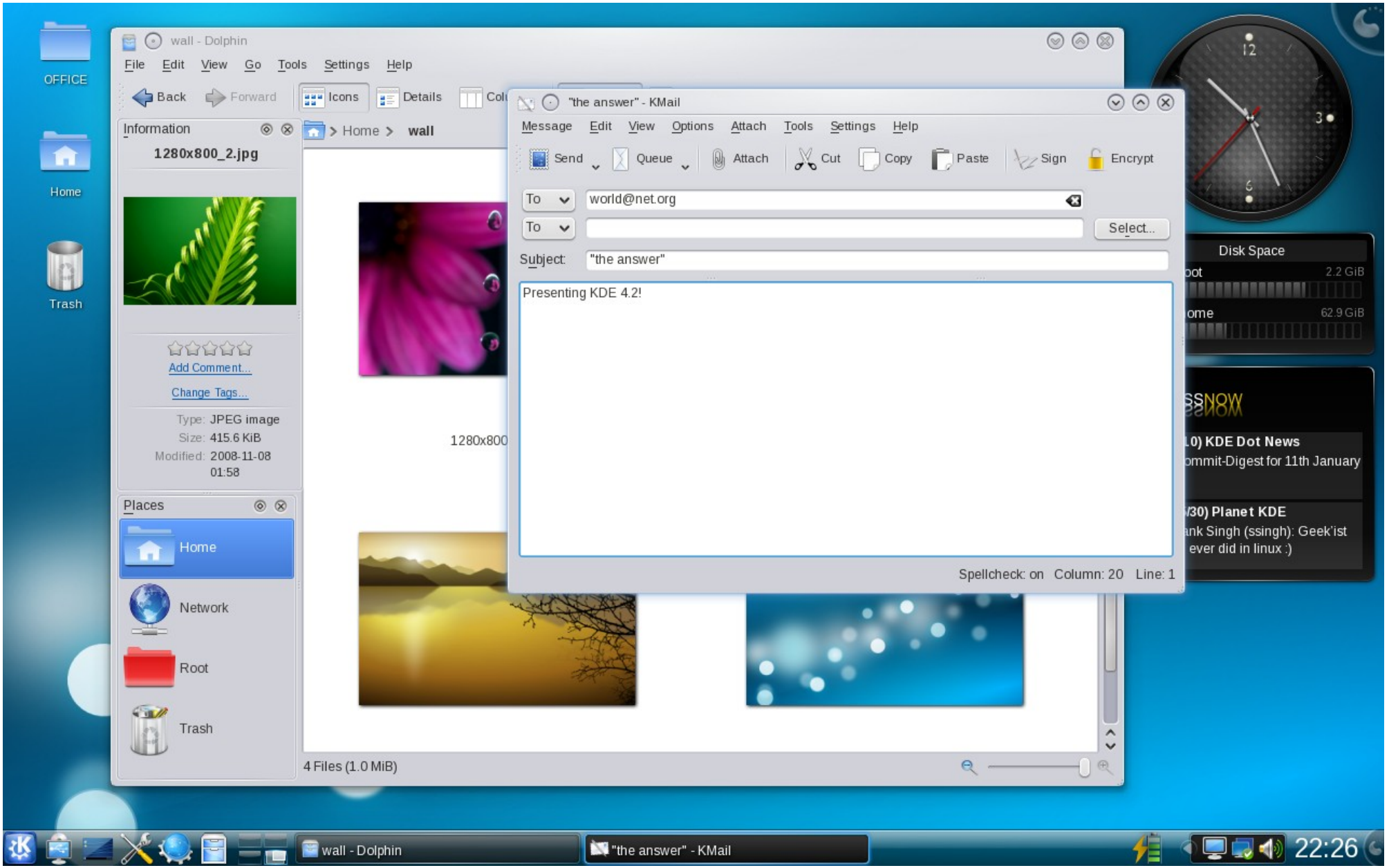
X with GNOME


Home directory
logo-red.xpm
redhat-logo-new.gif
GnomeCard
File Edit Help
New Open Save Add Edit Del First
Card Name Email Organization
Dave
drmike drmike@redhat.com RHAD Labs
marc marc@redhat.com RHAD Labs
raster raster@redhat.com RHAD Labs
/home/demo/Spread Sheets
File Window Commands Options Desktop Help
Location: /home/demo/Spread Sheets
.automount
bin
boot
cdrom
dev
etc
floppy
gnome
gnome-old
golem
home
build
demo
.cedit
.e-conf
.ee
.enlightenment
.gnome
.gnome_private
.mc
.netscape
.ssh
.themes
Contact
E
GRAPHICS
Textures
gnome
gnome-screenshots
logos
misc
Images
Mail
My Documents
My Sounds
Spread Sheets
desktop
dork
ftp
Hypothetical-sales.gnumeric
test-date.gnumeric
test-indent.gnumeric
test-math.gnumeric
1,259 bytes in 1 file
Control Center
Main Help
Multimedia
Appearances
Look and Feel
Theme Selector
Workspace
Input Devices
Theme Selector
Available Themes
Cheese
Default
Forest
Metal
Pixmap
ThinIce
Win95
Preview
Selected themes from above will be tested by previewing here.
Sample Button
Sample Check Button
Sample Text Entry Field
Radio Button 1
Radio Button 2
Radio Button 3
One
Two
Eenie
Meenie
Mynie
Moe
Catcha
Tiger
Try
Revert
OK
Cancel
Help
Select which desktop theme to use.
Gnumeric: Hypothetical-sales.gnumeric
File Edit View Insert Format Help
A1
DOS
Linux
SCO
January 1,000 900 500
February 900 2,500 300
March 600 4,100 100
April 700 5,700 (100)
May 600 7,300 (300) Bad year
June 500 8,900 (500)
July 400 10,500 (700)
August 300 12,100 (900)
September 200 13,700 (1,100)
October 100 15,300 (1,300)
November 0 16,900 (1,500)
December (100) 18,500 (1,700)
Totals: 5,400 116,400 (7,200)
Zoom out
Zoom in
Sheet 0
Info
SUM=0
/home/demo/S...
Netscape: GN...
GnomeCard
Control Center
Gnumeric: hyp...
demo@majesti...
Print
Fri Dec 11
03:30 PM

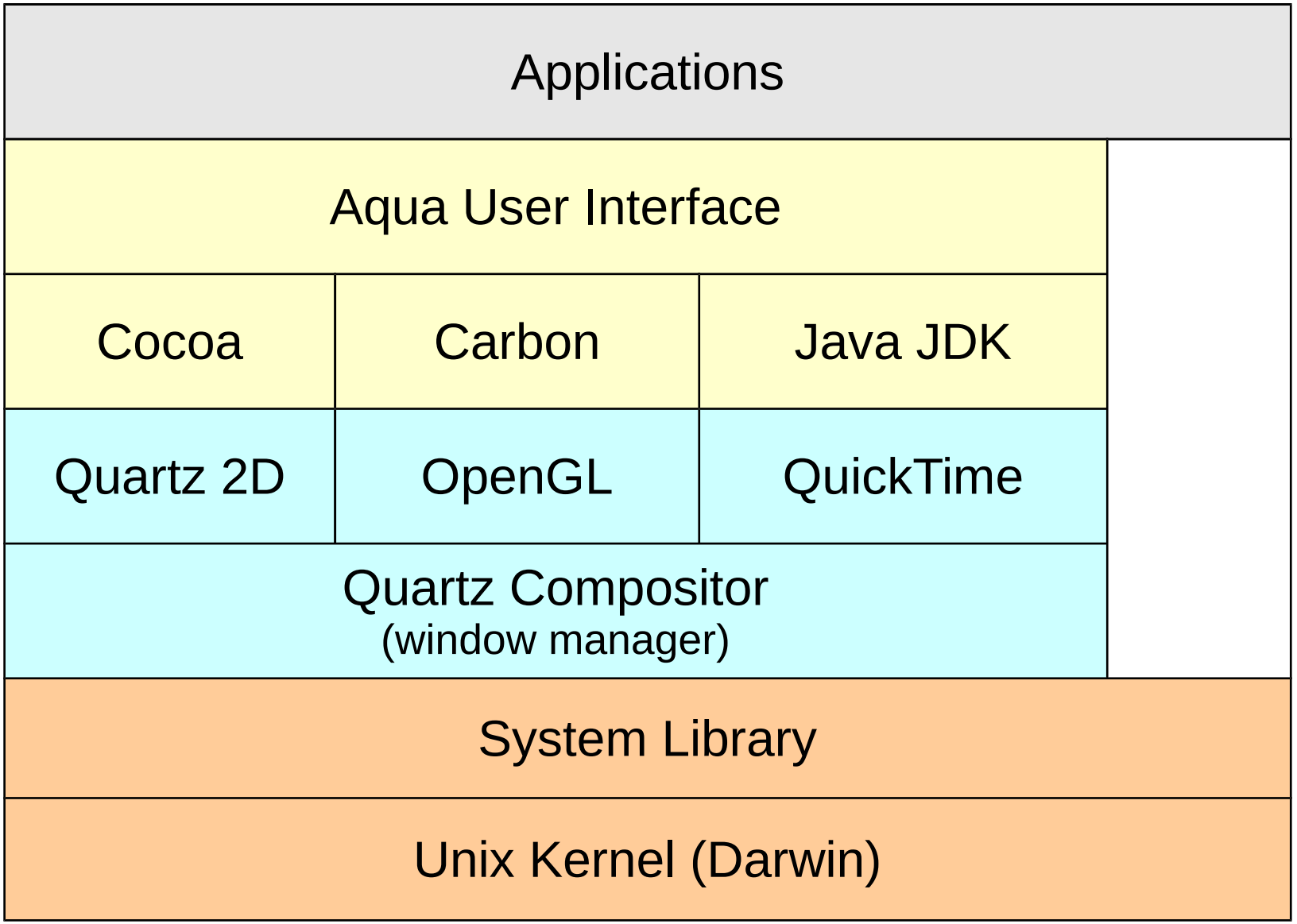
X with KDE


Applications
Toolkit
K Desktop Environment
Xt Intrinsics
Xlib
X Protocol
System Library
UNIX / Linux Kernel

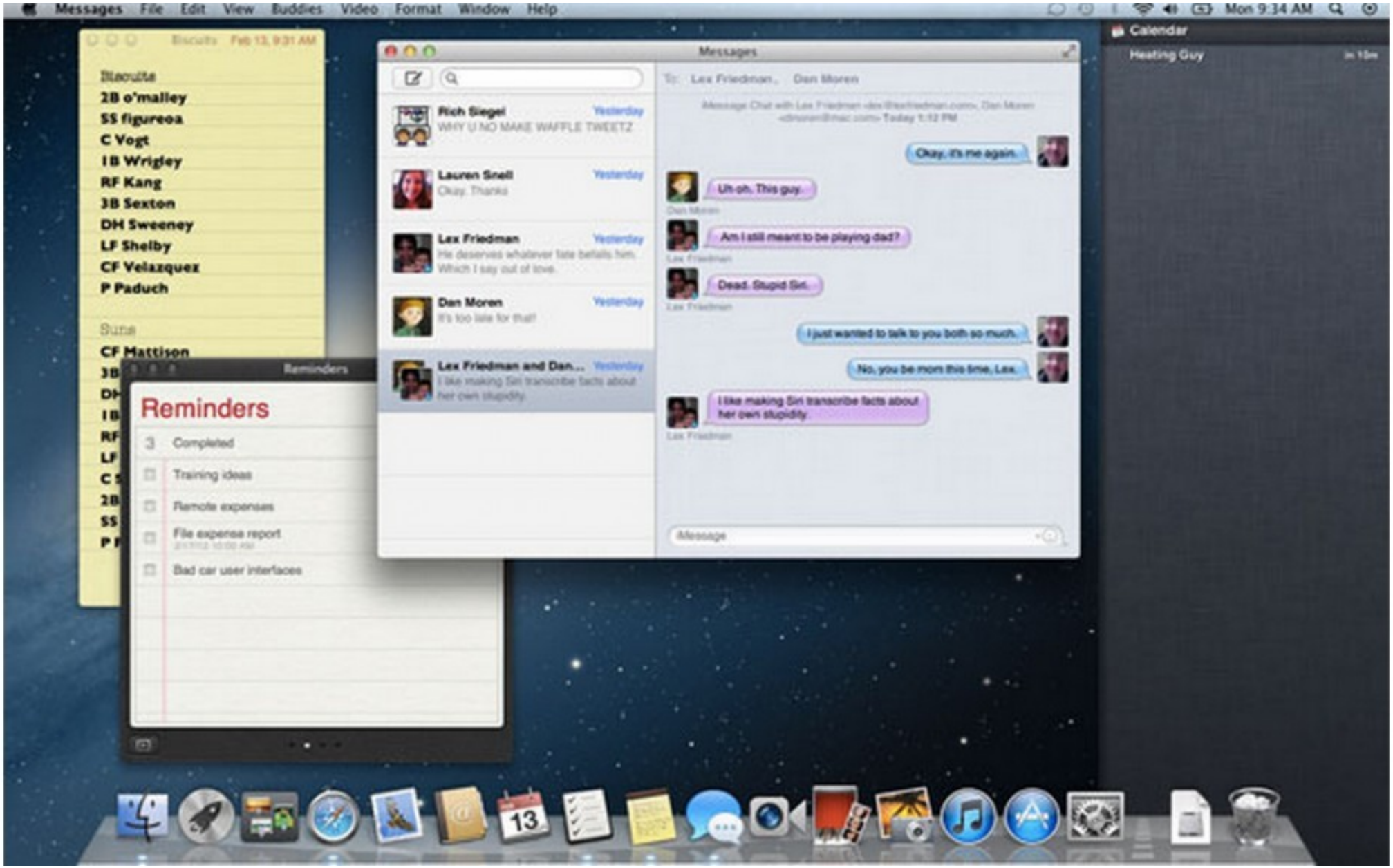
X with KDE


wall - Dolphin
File Edit View Go Tools Settings Help
Back Forward Icons Details
Information
1280x800_2.jpg
Add Comment...
Change Tags...
Type: JPEG image
Size: 415.6 KiB
Modified: 2008-11-08 01:58
Home > wall
Places
Home
Network
Root
Trash
4 Files (1.0 MiB)
"the answer" - KMail
Message Edit View Options Attach Tools Settings Help
Send Queue Attach Cut Copy Paste Sign Encrypt
To world@net.org
To
Select...
Subject: "the answer"
Presenting KDE 4.2!
Spellcheck: on Column: 20 Line: 1
Disk Space
2.2 GiB
62.9 GiB
KDE Dot News
Commit-Digest for 11th January
Planet KDE
Frank Singh (ssingh): Geek'ist ever did in linux :)
OFFICE
Home
Trash
wall - Dolphin
"the answer" - KMail
22:26

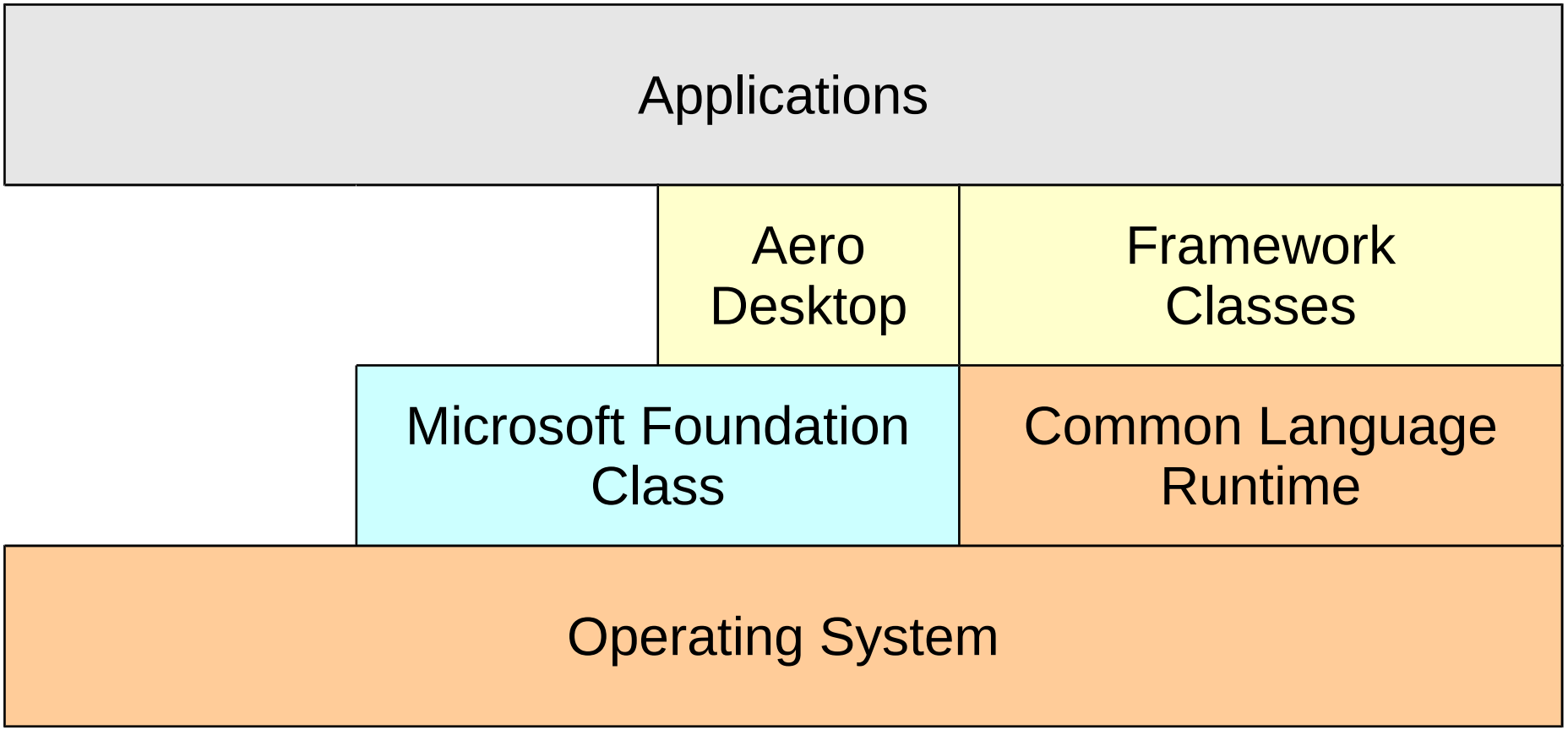
Mac OS X


Applications
Aqua User Interface
Cocoa
Carbon
Java JDK
Quartz 2D
OpenGL
QuickTime
Quartz Compositor
(window manager)
System Library
Unix Kernel (Darwin)

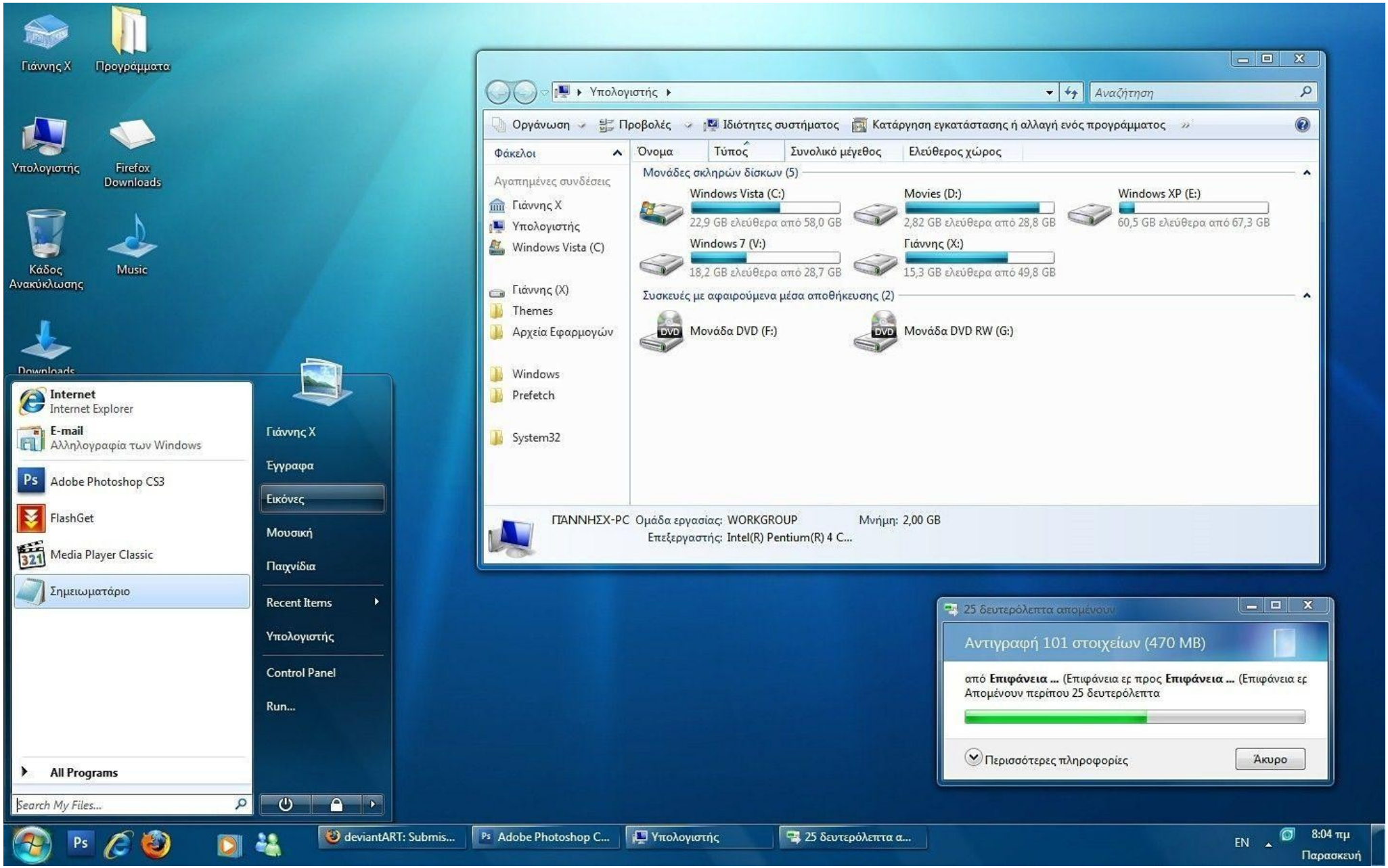
Mac OS X Mountain Lion



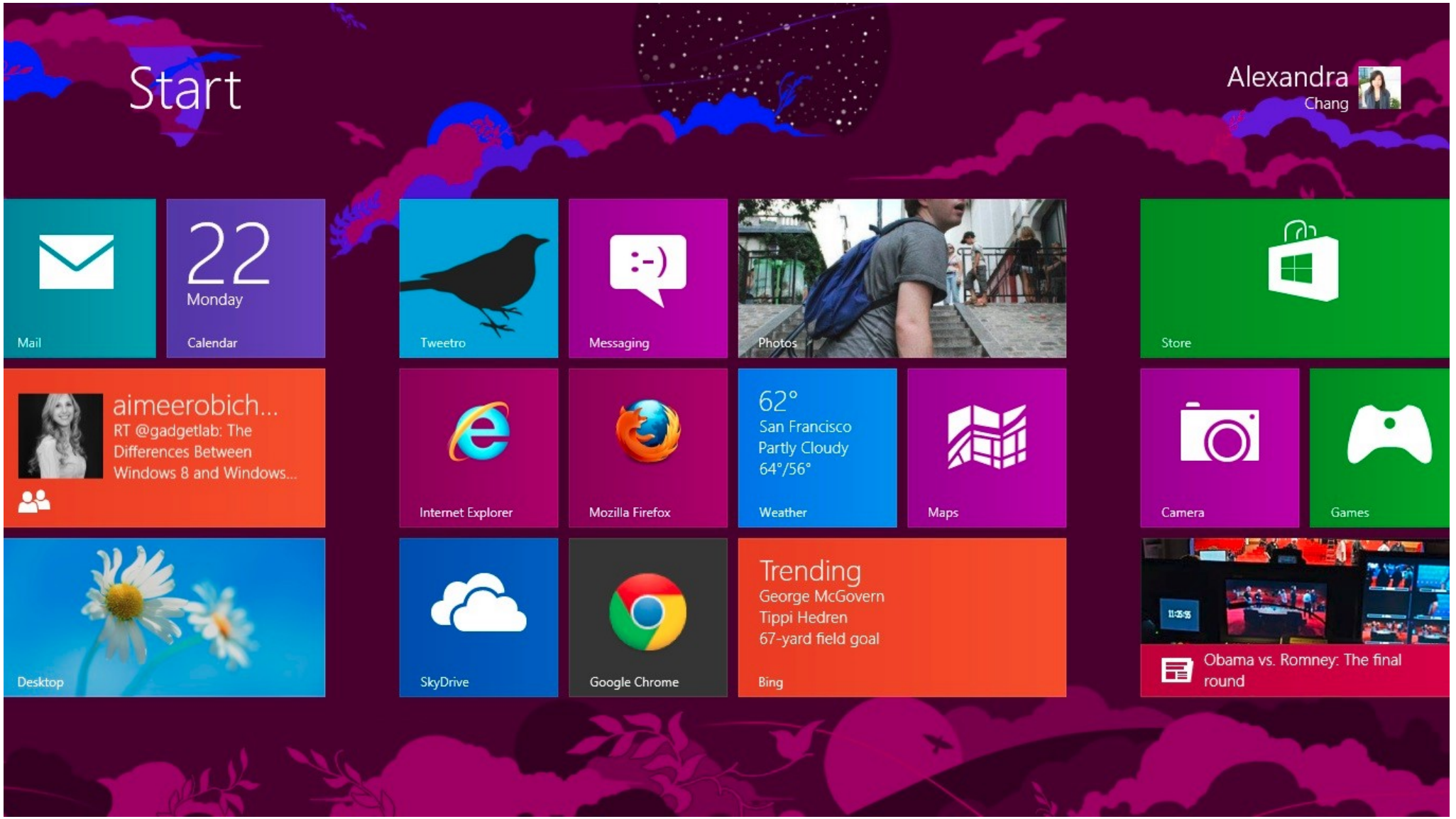
Microsoft Windows



Windows 7

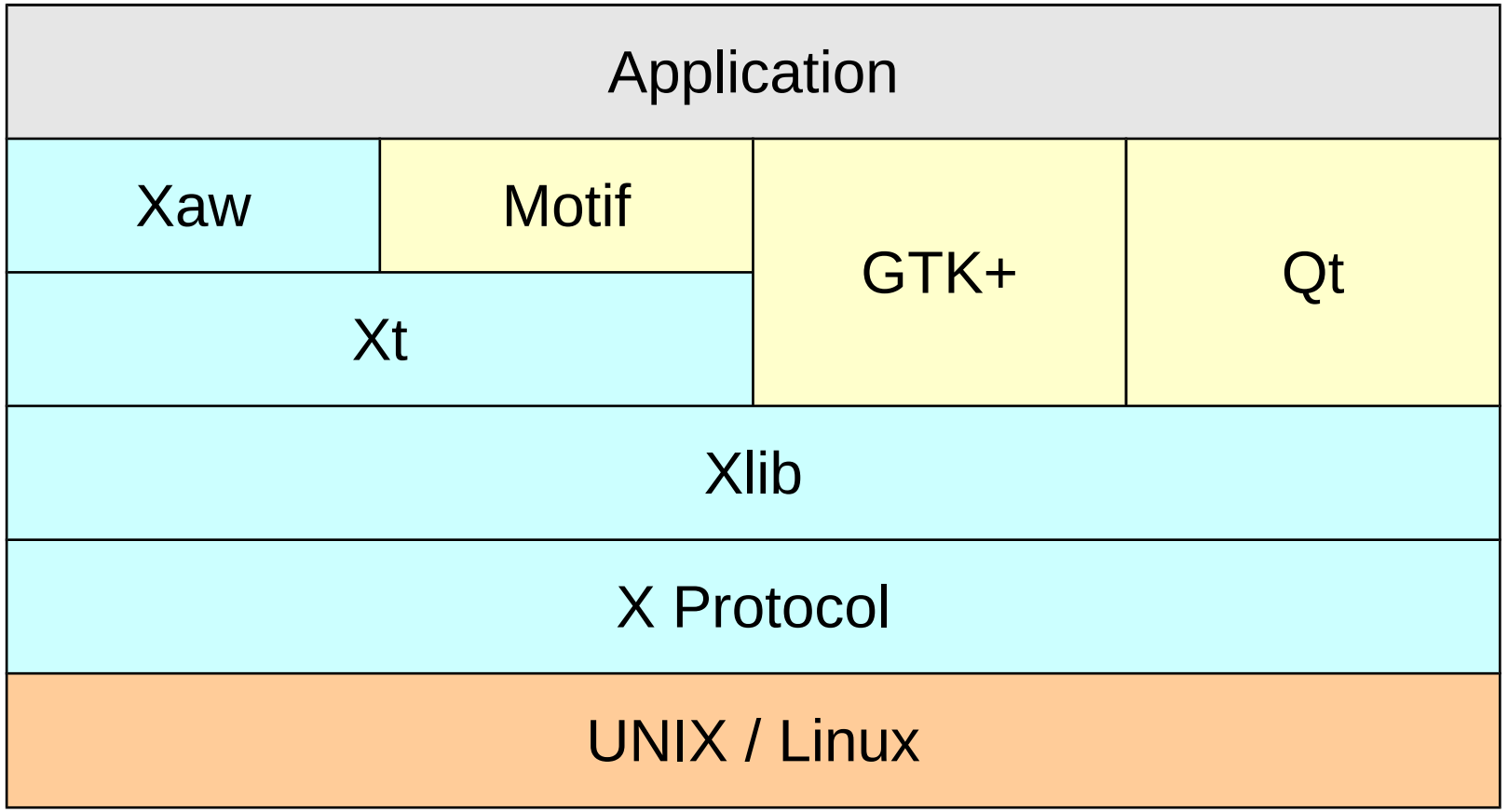

Γιάννης Χ
Προγράμματα
Υπολογιστής
Firefox Downloads
Κάδος Ανακύκλωσης
Music
Downloads
Internet
Internet Explorer
E-mail
Αλληλογραφία των Windows
Adobe Photoshop CS3
FlashGet
Media Player Classic
Σημειωματάριο
All Programs
Search My Files...
Γιάννης Χ
Έγγραφα
Εικόνες
Μουσική
Παιχνίδια
Recent Items
Υπολογιστής
Control Panel
Run...
Υπολογιστής
Αναζήτηση
Οργάνωση
Προβολές
Ιδιότητες συστήματος
Κατάργηση εγκατάστασης ή αλλαγή ενός προγράμματος
Φάκελοι
Όνομα
Τύπος
Συνολικό μέγεθος
Ελεύθερος χώρος
Αγαπημένες συνδέσεις
Γιάννης Χ
Υπολογιστής
Windows Vista (C)
Γιάννης (X)
Themes
Αρχεία Εφαρμογών
Windows
Prefetch
System32
Μονάδες σκληρών δίσκων (5)
Windows Vista (C:)
22,9 GB ελεύθερα από 58,0 GB
Movies (D:)
2,82 GB ελεύθερα από 28,8 GB
Windows XP (E:)
60,5 GB ελεύθερα από 67,3 GB
Windows 7 (V:)
18,2 GB ελεύθερα από 28,7 GB
Γιάννης (X:)
15,3 GB ελεύθερα από 49,8 GB
Συσκευές με αφαιρούμενα μέσα αποθήκευσης (2)
Μονάδα DVD (F:)
Μονάδα DVD RW (G:)
ΓΙΑΝΝΗΣΧ-PC Ομάδα εργασίας: WORKGROUP Μνήμη: 2,00 GB
Επεξεργαστής: Intel(R) Pentium(R) 4 C...
25 δευτερόλεπτα απομένουν
Αντιγραφή 101 στοιχείων (470 MB)
από Επιφάνεια ... (Επιφάνεια εε προς Επιφάνεια ... (Επιφάνεια εε
Απομένουν περίπου 25 δευτερόλεπτα
Περισσότερες πληροφορίες
Άκυρο
deviantART: Submis...
Adobe Photoshop C...
Υπολογιστής
25 δευτερόλεπτα α...
EN
8:04 πμ
Παρασκευή

Windows 8


Start
Alexandra
Chang
Mail
22
Monday
Calendar
Tweetro
Messaging
Photos
Store
aimeerobich...
RT @gadgetlab: The Differences Between Windows 8 and Windows...
Internet Explorer
Mozilla Firefox
62°
San Francisco
Partly Cloudy
64°/56°
Weather
Maps
Camera
Games
Desktop
SkyDrive
Google Chrome
Trending
George McGovern
Tippi Hedren
67-yard field goal
Bing
Obama vs. Romney: The final round

Qt

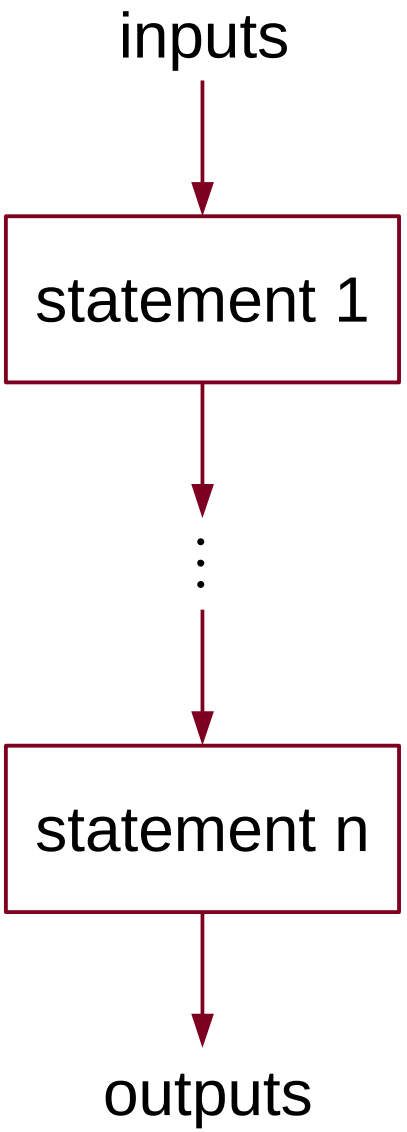
- In UNIX and Linux system, Qt sits on top of Xlib.



Xaw: Athena widgets

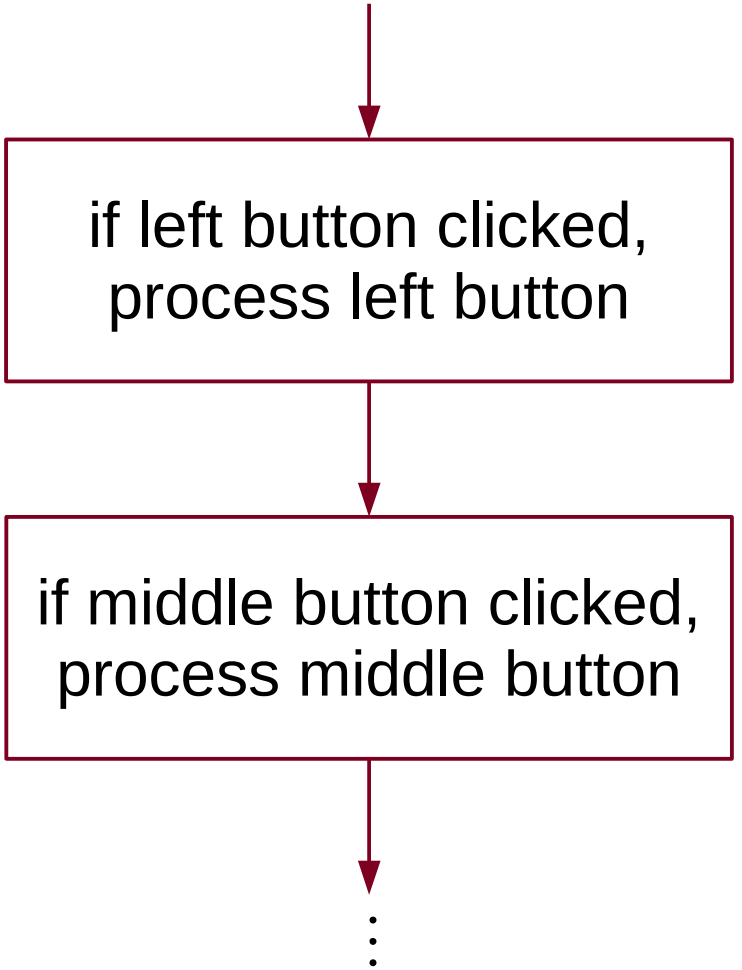
Sequential Program

- Sequential program runs one statement at a time.
- A statement is run till completion before proceeding to next statement.

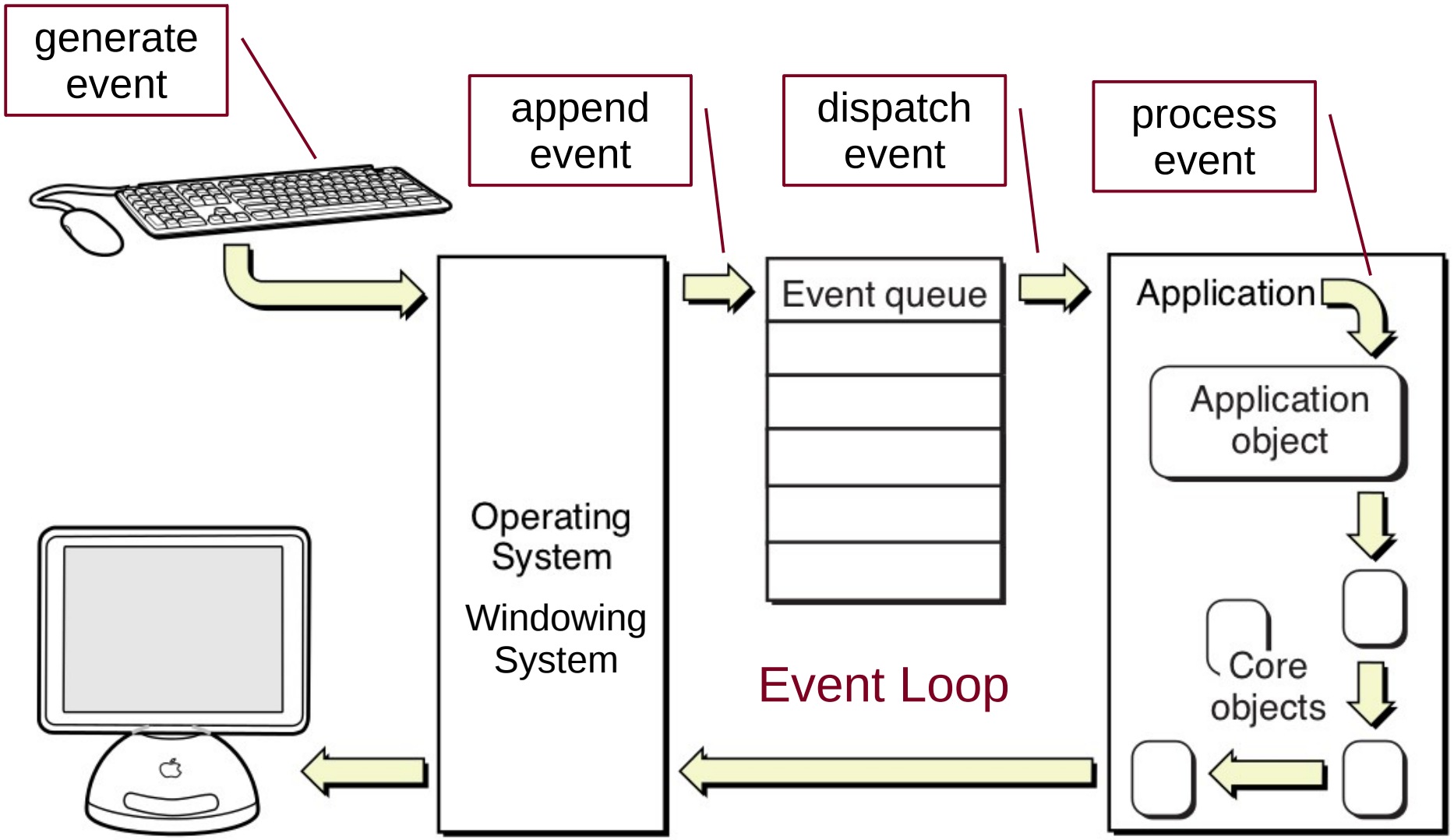


Polling

- Sequentially check for user actions.
- Slow if long check list.
- Can't respond to user actions promptly.



Event-Driven Program



- Events signal user actions, e.g.,
  - mouse click, mouse move
  - key press, key release
  - window activated, window deactivated
- Events can be generated by
  - operating system
  - device drivers
  - windowing system, window manager
  - application program
- Events tend to be too low-level
  - When left-mouse button is clicked, which item is selected?
- Windowing systems provide higher-level mechanisms.

X / Motif

- Implemented in C.
  - C++ hasn't appeared yet!
- Use C `struct` to emulate object class.
- Use callbacks: functions that perform GUI tasks.

```
XtCallbackProc callback(widget, data, callback_data)
```

- Callbacks are attached to widgets.

```
XtAddCallback(widget, event_type, callback, data)
```

- Widgets call callbacks when activated.
  - Widgets generate `callback_data` according to `event_type`.

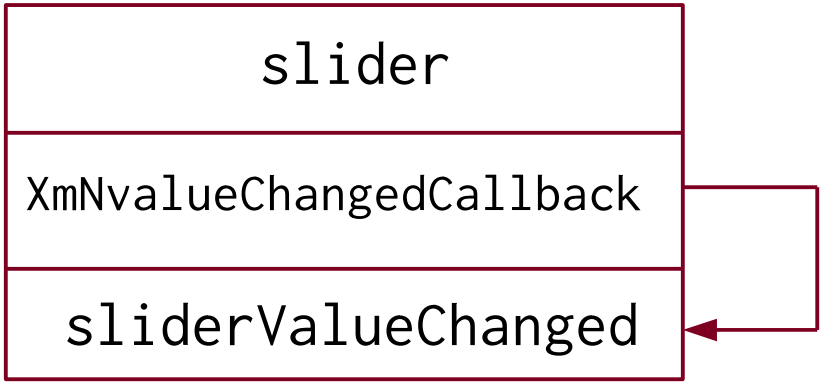
```
// Create widgets
slider = XmCreateScale(parent, “slider”, ...);
spinBoxText = XmCreateTextField(parent, “text”, ...);

// Add callbacks to widgets
XtAddCallback(slider, XmNvalueChangedCallback,
    sliderValueChanged, spinBoxText);
XtAddCallBack(spinBoxTex, XmNactivateCallback,
    spinBoxTextVerify, slider);

// Define callback functions
XtCallbackProc sliderValueChanged(Widget slider,
    Widget spinBoxText, XmScaleCallbackStruct *cbs)
{
    int value;
    XtVaGetValues(slider, XmNvalue, &value, 0);
    XtVaSetValues(textField, XmNposition, value, 0);
}
```

```
XtCallbackProc spinBoxTextVerify(Widget spinBoxText,
  Widget slider, XmAnyCallbackStruct *cbs)
{
  ...
  changed = 0;
  str = XmTextFieldGetString (textField);
  if (str && (ret = sscanf(str, "%d", &value)) > 0)
    if (value >= MinValue && value <= MaxValue) {
      XtVaSetValues(textField, XmNposition, value, 0);
      XtVaSetValues(slider, XmNvalue, value, 0);
      changed = 1;
    }
  ...

  if (!changed) {
    XtVaGetValues(textField, XmNposition, &value, 0);
    sprintf(string, "%d", value);
    XmTextFieldSetString(textField, string);
  }
}
```



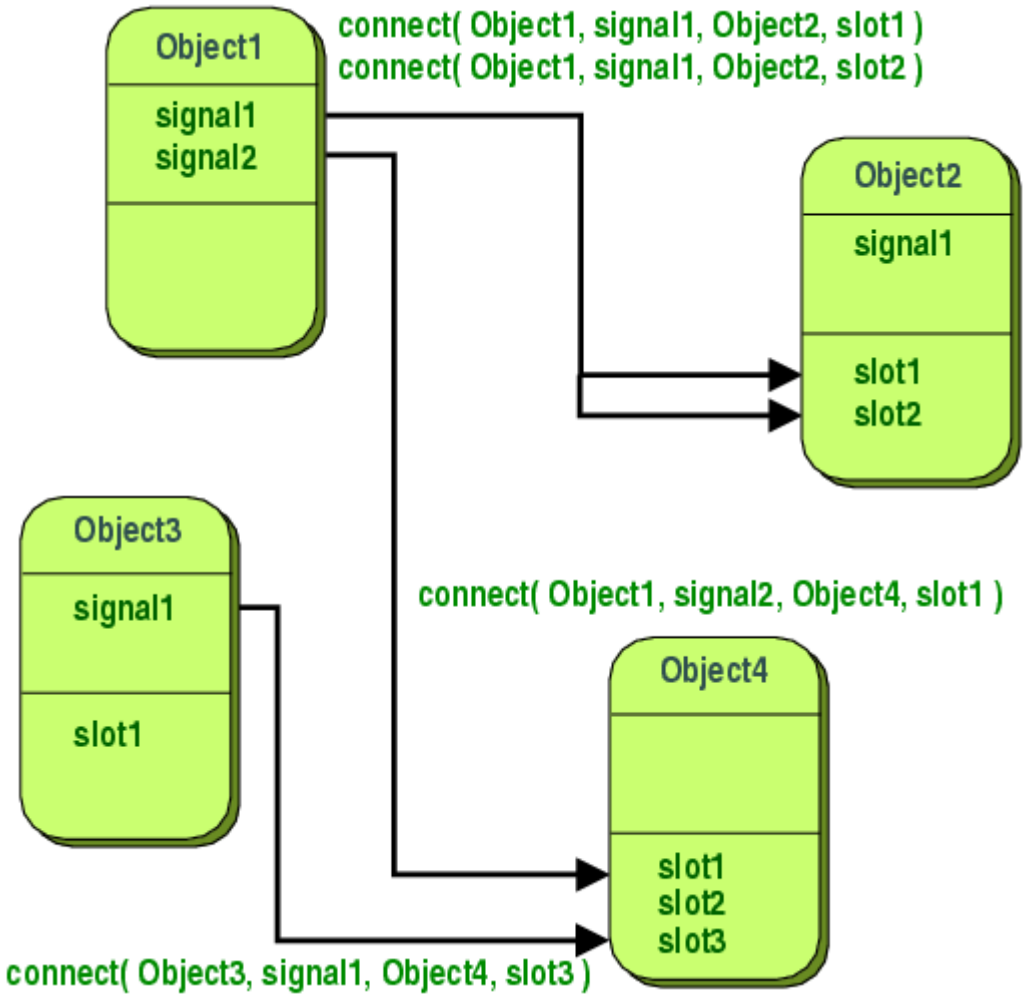
- Can think of callbacks as belonging to widget.
  But, programmer implements the callbacks.
- `slider`'s `XmNvalueChangedCallback` event is linked to `sliderValueChanged` callback function.
- After `slider` is dragged to new position, it inserts `XmNvalueChangedCallback` event into event queue.
- X event dispatcher processes event and
  calls `sliderValueChanged`.

Callbacks have two shortcomings

- Not type-safe:
  - All callbacks have the same generic arguments.
  - Cannot guarantee calling function passes correct arguments.
- Tight coupling:
  - Widgets are explicitly linked to callbacks.

Qt

- Implemented in C++.
- Adopts signals and slots.
- Object emits signal in response to event.
- Object has slots that perform tasks.
- Connect signals to slots in application.



- Can connect one signal to multiple slots.

```
connect(slider, SIGNAL(valuedChanged(int)),
   spinBox, SLOT(setValue(int)));
connect(slider, SIGNAL(valuedChanged(int)),
   this, SLOT(updateStatusBar(int)));
```

- Can connect multiple signals to same slot.

```
connect(lcd, SIGNAL(overflow()),
   this, SLOT(handleMathError()));
connect(calculator, SIGNAL(divisionByZero()),
   this, SLOT(handleMathError()));
```

- Can connect signal to signal.

```
connect(lineEdit, SIGNAL(textChanged(const QString &)),
   this, SIGNAL(updateRecord(const QString &)));
```

Advantages of signals and slots

- Loose coupling:
  - Objects emit signals in response to events.
  - Objects are not explicitly linked to slot functions.
- Type safe:
  - Arguments of signals and slots must match.
- Flexible:
  - Not limited to widgets,
    can be used by any `QObject` subclass.

Implemented by meta-object system.

Microsoft Windows

Three ways to create Windows applications in C++

- Use Windows API
  - Calls Windows OS functions. Low-level programming.
- Use Microsoft Foundation Class (MFC)
  - Encapsulates Windows API.
  - Higher-level programming.
- Use Windows Forms
  - For forms-based application that run with Common Language Runtime.

MFC

- MFC provides two mechanisms
  - Message mapping
  - Data binding
- Message Mapping
  - Messages
    - Like higher-level events.
  - Message handlers
    - Like callback functions.
  - Message map
    - Connects messages to message handlers.

- Message map handles 4 kinds of messages:
  - WM_COMMAND
    - menu
  - Child window messages
    - button, combo box, text edit, list box
  - WM_Message
    - miscellaneous window messages
  - User defined messages

- Why divide into 4 categories? Why not just 1?

```
class CAgeDialogApp: public CWinApp
{
    ...
    CSliderCtrl slider;
    CEdit spinBoxText;
    ...
    afx_msg void OnHScroll(UNIT code, UNIT position,
        CScrollBar *scrollBar);
    afx_msg void OnEnUpdate();
    DECLARE_MESSAGE_MAP()
};
```

- `afx_msg` declares message handlers.
- `DECLARE_MESSAGE_MAP` declares that the class contains message handlers.

```
// In .c file
BEGIN_MESSAGE_MAP(CAgeDialogApp, CWinApp)
    ON_WM_HSCROLL()  // maps to OnHScroll
    ON_EN_UPDATE(IDC_SPINBOXTEXT,
        &CAgeDialogApp::OnEnUpdateTextBox)
END_MESSAGE_MAP()
...


void CAgeDialogApp::OnHScroll(UNIT code, UNIT position,
    CScrollBar *scrollBar)
{
    ... // update text box's state
}

void CAgeDialogApp::OnEnUpdateTextBox()
{
    ... // update slider's state
}
```

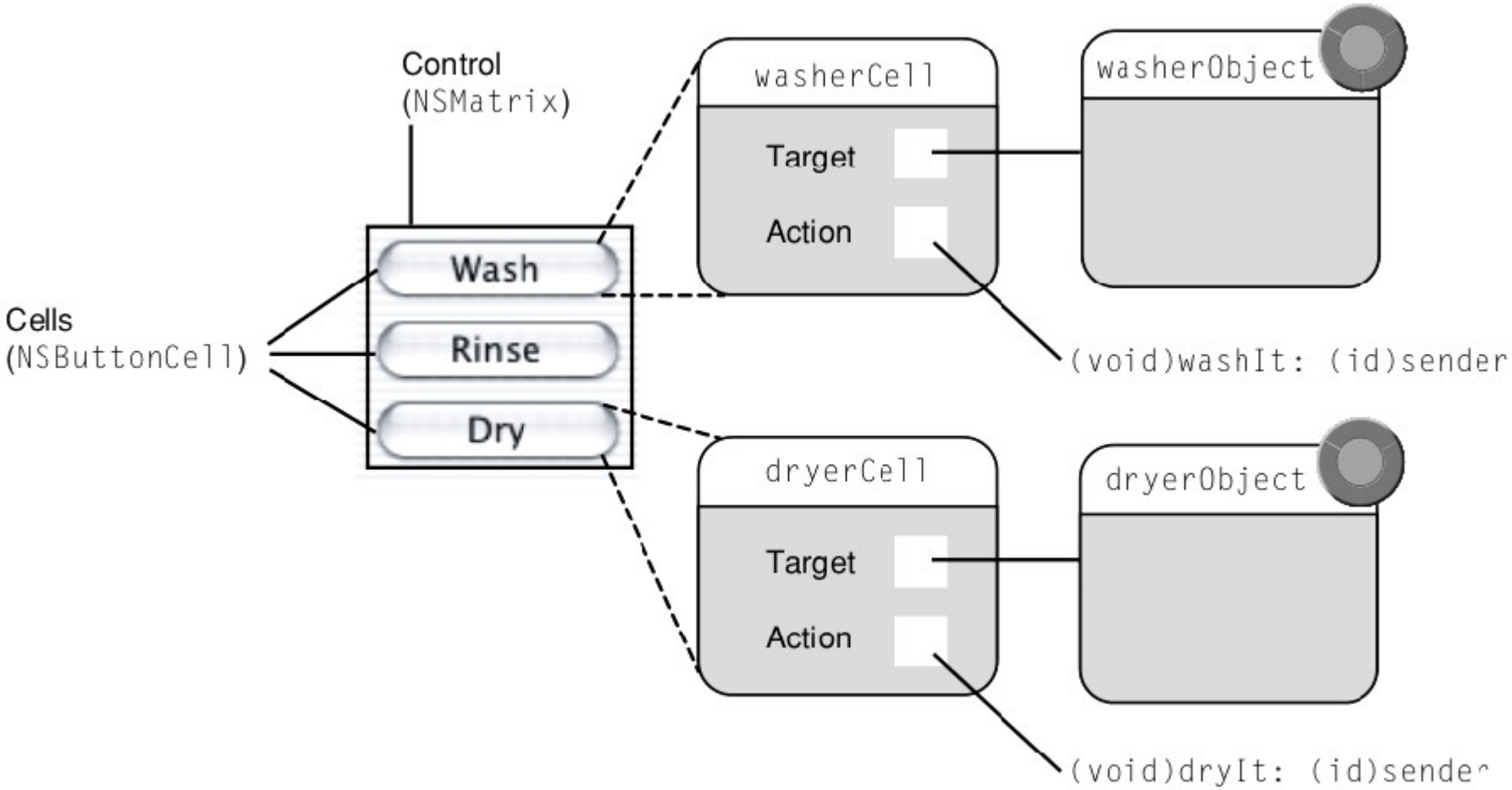
Mac OS X

- Provides several tools
  - Cocoa
  - Carbon
- Cocoa is implemented in Objective-C.
- Cocoa provides two mechanisms:
  - Outlets and Actions
  - Bindings
    - Discussed in *Model-View Patterns*.

Outlets and Actions

- Cocoa widgets are called controls and cells.
- Cells have outlets that point to target objects.
- In response to a user action,
  cell generates action message.
- Action message is sent to target,
  i.e., call target's action method (function).

Cocoa target-action mechanism



- Similar to callback mechanism
  - action message ≈ widget event
  - action method ≈ callback

```
// In .h file

@property (assign) IBOutlet NSSlider *slider;
@property (assign) IBOutlet NSTextField *spinBoxText;


// In .m file

- (IBAction)sliderValueChanged:(NSSlider *)sender {
    [[self spinBoxText] setIntegerValue:
      [sender integerValue]];
}

- (IBAction)spinBoxTextValueChanged:(NSTextField *)sender {
    [[self slider] setIntegerValue:
      [sender integerValue]];
}
```

- Connections between outlets and actions are done in UI builder.

Comparisons

- X Window callbacks
  - Use C functions to connect widgets to callbacks.
  - Programmer writes callbacks.
  - Programmer writes connection codes.
  - Simple and easy to understand, but troublesome to write.

- MFC message map
  - Use C++ macros to create codes for message map.
  - Programmers write message handlers, like callbacks.
  - Programmer inserts macros to make connections.
  - Easy to understand at conceptual level, but tend to be messy.

- Cocoa outlets and actions
  - Use Objective-C extensions to implement outlets and actions.
  - Use UI builder to connect outlets, actions and target.
  - Programmer writes action codes (like callbacks), no need to write connection codes.
  - Simple but programming language mechanism is obscured.

- Qt signals and slots
  - Use C++ functions to implement meta-object system.
  - Use meta-object compiler qmake to auto create codes.
  - Widgets come with slot functions.
  - Programmer writes simple connection codes.
  - Easy to understand at conceptual level.

Summary

- GUI works on top of windowing system.
- GUI programs are event-driven.
- Windowing system provides event-handing and higher-level mechanisms.
- X Window System uses callbacks.
- Qt uses signals and slots.
- Microsoft Windows uses message map.
- Mac OS X uses outlets, actions and bindings.

Further Readings

- GUI Comparisons: attached document.
- Signals and slots: [Blan2008] p. 20–22.
- X Window System: [Kell90] chap. 1.
- Motif spin box and callback: [Foun2001] chap. 15.
- MS Windows: [Hort10] p. 1–8.
- MFC message map: [Hort10] p. 903–910.
- Mac OS X: [Tren10] p. 4–20.
- Cocoa outlets, actions, bindings: [Tren10] chap. 8, [Cocoa] p. 217–227.

References

- *Cocoa Fundamentals Guide*, Apple Inc., 2010.
- J. Blanchette and M. Summerfield, *C++ GUI Programming with Qt 4*, 2nd ed., Prentice Hall, 2008.
- A. Fountain, J. Huxtable, P. Ferguson and D. Heller, *Motif Programming Manual*, O'Reilly, 2001.
- I. Horton, *Beginning Visual C++ 2010*, Wiley, 2010.
- B. J. Keller, *A Practical Guide to X Window Programming*, CRC, 1990.
- M. Trent and D. McCormack, *Beginning Mac OS X Snow Leopard Programming*, Wiley, 2010.