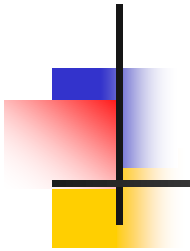


System Programming

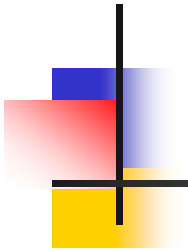
Unix Window System

Why Window Systems?

- A window system provides a graphical user interface (GUI) based on windows, icons, and interrupt-driven interaction
- Increased usability due to
  - Access to multiple environments and applications at once
  - Direct manipulation of graphical objects with mice, joystick, tablets, etc.

Window Systems and Unix

- Unix evolved before window systems and optimized use of the command line
- Modern Unix systems include a window system to combine the advantages of the window system with the availability of a command line for expert use (that means you!)



X Windows

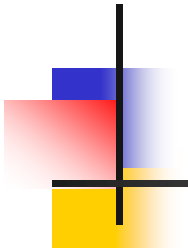
- Practically all Unix window systems are based on X Windows (XFree86)
- Standard Version: X11R6
- X Server:
  - Hardware interface (display, mouse, etc.)
  - Manages the screen space
  - Draws simple graphics
  - Assigns rectangular regions to X clients
  - Local and remote clients supported

The X Client-Server Architecture

- **X** is actually designed to work over a network
- **X server:** software that runs on the machine where the program's output will be displayed
- **X client:** program running on the same or another machine
- Client sends drawing and other X commands to the server, which displays the results

Historical Use of X

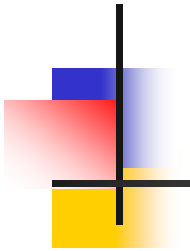
- Users sat at an X terminal – graphical terminals that ran X server, but no OS
- User logged into remote computer running UNIX or other OS supporting X clients
- Separates graphical interfaces and manipulates from application
- Combine applications running on multiple computers



Features of X

- Transparent remote execution
- Gives each program its own virtual screen
- Includes important windowing concepts
  - Window damage
  - Window reveal events
  - Backing store
- X11 programs are highly portable

Window Manager



- Window manager runs on top of X11 and within a desktop manager
- Place borders, sliders, and other widgets on windows to provide the interface look and feel
- Examples:
  - kwin – default for KDE
  - metacity – default for GNOME
  - mwm – Motif standalone window manager

Desktop Environment

- Desktop environment organizes display into an integrated environment
- Includes file managers (Nautilus), icons, panels, configuration tools, and applets
- GNOME (GNU)
  - Built on GIMP Toolkit (GTK+)
- KDE (uses Qt C++ libraries)
- Xfce (GTK+ based)