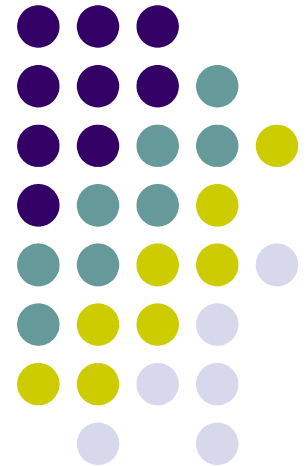
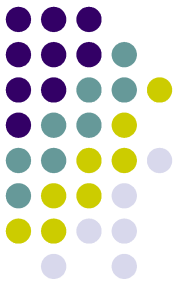


X Window Manager and Remote Displays

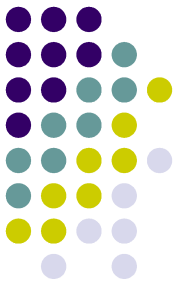
History and usage





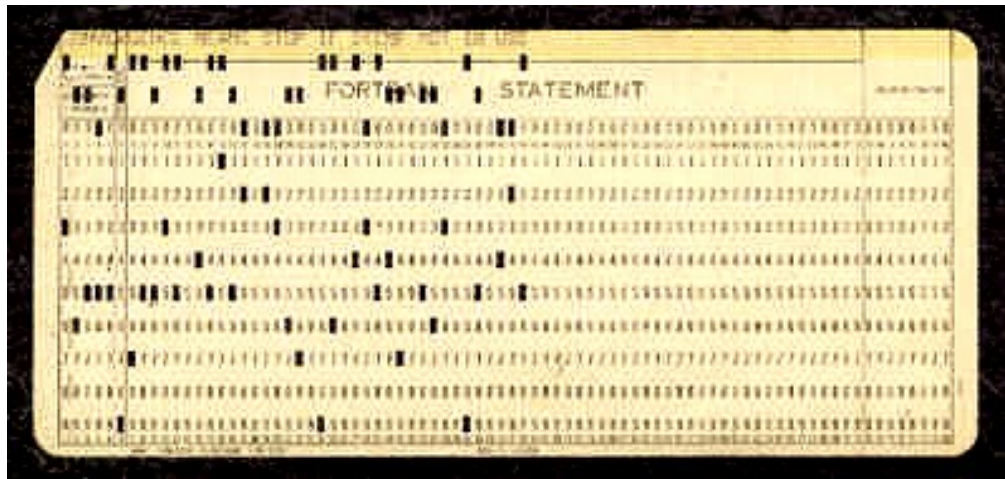
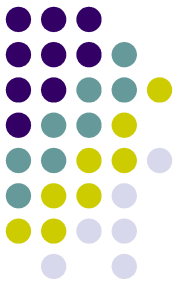
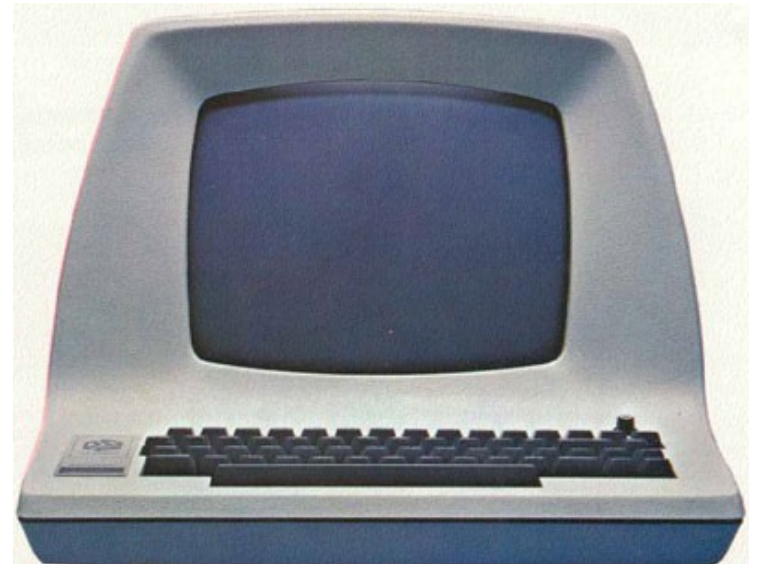
Interface history

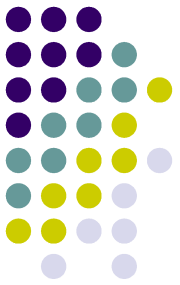
- 1) Punch Cards and Dip Switches
- 2) Teletype and Glass teletype (output only)
- 3) Dumb Terminals (limited cursor)
- 4) Terminals w/ escape sequences(vt100)
- 5) VT220 w/ ReGIS
- 6) X Terminals



The beginning

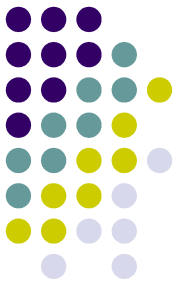
- Punch Cards and dipswitches are the only way to interact with mainframes
- Eventually mainframes begin to include a single glass teletype, which displays text like error messages and job notifications
- Finally terminals are connect by serial ports with multiple terminals being connected at the same time

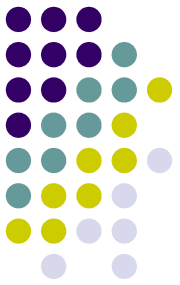




DEC sets the standard

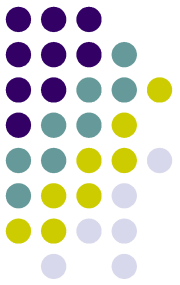
- The year is 1978
- DEC introduces the VT100 which supports the ANSI standard X3.64 (ASCII)
- Uses a standard intel 8080 processor to handle the command processing
- Basic text input/output and with limited cursor movement. Putty is a vt100 emulator.





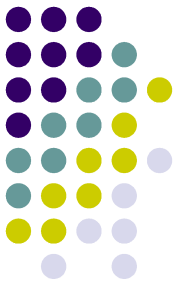
W and X

- Developers at MIT needed a graphical system that could work across multiple platforms
- The existing windowing System Was called W
- X version 1 was completed in 1984
- Rapid development cycle led to X version 6 release in 1985
- X version 10 is the first to be widely deployed



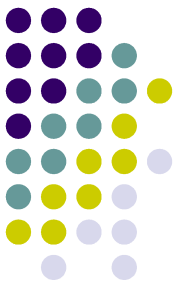
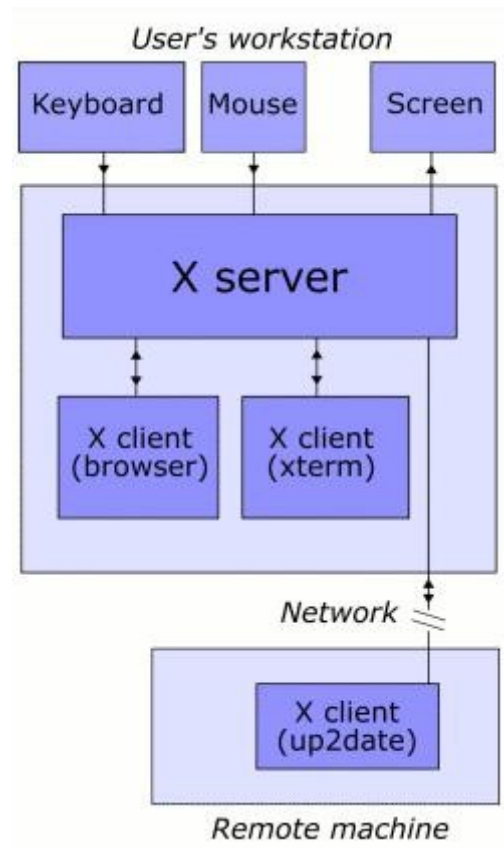
X11 and Xfree86

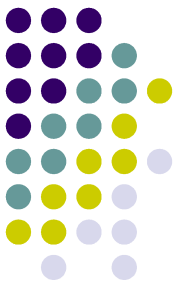
- Originated in 1992 from X386 server
- Quickly became the de-facto standard for intel machines
- The current Linux implementation is based off Xfree86
- Complies with the same standards as other X clients for Mac OSX, Solaris, Aix, BSD, etc.



X Model

- X is a client server model
- Most machines actually have three processes running
- XDM is the X Display Manager
- Xserver which manages all the clients
- And X which is the main X client and spawns are child clients
- Finally the window manager is running on top of X





```
# the requestor can now read the selection in the property of the window by sending one or more <code>GetProperty</code> requests to the server;
# the requestor destroys the property; if the selection owner has requested to be informed of this, it is send a <code>PropertyNotify</code> event.
```

This is the sequence of steps that take place when the content of the selection is considered sufficiently small to be transferred all at once. If the content is large, it should be transferred in chunks. In this case, both clients express interest in <code>PropertyNotify</code> events; this way, the selection owner knows when the selection has been read, and the requestor knows when another chunk has been placed in the property.

```
==Cut buffers==
```

Data, and in particular the selected text, can also be transferred using cut buffers. These are simply properties of the [[root window]], named <code>CUT_BUFFER</code>, etc. Contrarily to selections, the cut buffers do not involve a direct interaction between clients. Rather, when text is selected in a window, the window owner copies this text into the property of the root window of name <code>CUT_BUFFER</code>. When the user pastes the text in another window, the window owner reads this property of the root window.

```
==Xnd==
"x" 57L, 6881C
```

34,28 63%

```
hobbes{~}: [
```

Free Download

Firefox 1.0.7 for Linux i686, English (8.2MB)
[Other Systems and Languages](#)



Get Thunderbird

The [Thunderbird](#) e-mail client makes emailing safer, faster, and easier than ever before.



Mozilla Store

Get [Firefox](#) and [Thunderbird](#) on CD, the [official guidebook](#), t-shirts, and more at the new Mozilla Store.



Firefox 1.5 Release Candidate 3



Firefox 1.5
RC3

Web and Extension developers: join the

