

ARPANET

The Department of Defense sponsored research into building a reliable packet-switched network

This became the ARPANET and eventually the internet

The original network model was to have several nodes and links connecting them totally under control of the network control group

Clients wishing to connect to the network had to build a bit-serial interface device to their computer and plug the other end into one of the network nodes

The original protocol on the ARPANET guaranteed delivery of packets both between clients and between nodes

This resulted in much net traffic when a node went down and several undeliverable packets chased possible routes to the downed node until the slower protocol discovered that the node was unreachable

The original routing algorithm did not need global information

Each node kept a table, for each link, how long it would take a packet to be delivered to each node of the network

It then constructed a table of best routes and how good they were for each node

It then sent this table to each of its neighbors who could then update their tables

Of course, the route to one's self was zero and this made the system work

Eventually a centrally controlled routing strategy was developed

Xerox Contributions - Xerox PARC

Started in 1970 with the mission to create "the architecture of information"

In 1972 the first version of Smalltalk is deployed

In 1973 the first laser printer, called EARS (for Ethernet-Alto research character generator scanning laser output terminal) is in service, printing documents at 1 page/second at 500 spots per inch (spi)

A patent memo describing a new networking system uses the term "Ethernet" for the first time

The Alto personal computer becomes operational

The Alto will feature the world's first What-You-See-Is-What-You-Get (WYSIWYG) editor, a commercial mouse for input, a graphical user interface (GUI), and bit-mapped display, and will offer menus and icons, link to a local area network and store files simultaneously

The mouse was invented previously by Doug Englebart for the NLS system

Personal distributed computing is invented

A machine that runs just as fast during the day as at night

In 1974 PRESS, a Page Description Language, is developed

The Bravo word-processing program is completed, and work on Gypsy, the first bitmap What-You-See-Is-What-You-Get (WYSIWYG) cut & paste editor, begins. Bravo and Gypsy programs together represent the world's first user-friendly computer word-processing system.

BITBLT, an algorithm that enables programmers to manipulate images very rapidly without using special hardware, is invented. The computer command enables the quick manipulation of the pixels of an image and will make possible the development of such computer interfaces as overlapping screen windows and pop-up menus.

In 1975 engineers demonstrate a graphical user interface for a personal computer, including icons and the first use of pop-up menus. This interface will greatly influence the development of Windows and Macintosh interfaces

In 1980 the Interpress page description language is completed

Eventually, upset with the inappropriate actions by Xerox management, several researchers leave Xerox and found Adobe and create Postscript

In the mid '80s Xerox markets Lisp workstations that use the Interlisp-D programming language to support artificial intelligence programming as well as applications utilized within Xerox

The Data General NOVA Computer

When Digital Equipment Corporation was deciding the next computer to make there were 5 contenders

The winner was the pdp-11 that eventually became the VAX

A group decided that they had designed a good machine left the company and formed Data General and produced the NOVA line of computers

Characteristics

16-bit word length - 32768 word memory

Instructions had an 8-bit displacement

index 00 - the low 256 words of memory

index 01 - relative to PC

index 10 - relative to contents of ACC 2

index 11 - relative to contents of ACC 3

An indirect bit allowed indirect addressing

Certain words were "autoindexing" and would automatically increment if referenced as an indirect address

The Data General NOVA Instruction Set

Instructions

JMP - jump

JSR - jump and set ACC 3 with PC+1

ISZ - increment and skip if result zero

DSZ - decrement and skip if result zero

LDAn - load accumulator

STAn - store accumulator

IO - various IO instructions

Arithmetic and Logical instructions

Could specify

two accumulators (could be the same)

and a function

Complement

Negate

Move

Increment

Add Complement

Subtract

Add

And

the carry in

Carry

Complement of Carry

0

1

the shift

None

left

Right

Swap

a bit indicating if the result is only tested

and a Skip Command to skip if

No skip

Always skip

zero carry

Nonzero carry

zero result

nonzero result

skip either carry or result is zero

skip both carry and result nonzero

The Xerox ALTO

A microcoded computer

emulated a Data General NOVA computer

Used 16-bit address instead of Data General's 15-bit addresses

Did not allow multilevel indirect addresses

Did not implement auto increment locations

all IO performed by microcode

when it became time for the device to transfer a word the appropriate micro task was run and the word transferred from or to memory

Double word operations could be done in 1 microsecond for an IO bandwidth of up to 32 Mbits/second

Comparable to many mainframes of the day

Fast Task switcher

Microcode had a 16 word register of task program counters

Could switch tasks by selecting which word in register was used for program counter

IO service routines in microcode

rapid switching of micro tasks made individual word transfers possible for IO

Tasks for

disk reads

disk writes

bit mapped display

keyboard sensing

mouse tracking

NOVA emulator (lowest priority)

etc.

ALTO

Bit mapped display

scatter gather for each scan line

invert video

automatic curser display

Display task in microcode took up more than half of the CPU when displaying a full 606 x 808 pixels

Some programs would turn off much of the display to calculate faster

BCPL computer language used for all NOVA emulator programming for the system

Only datatype was the 16-bit word

How variables were used determined what they were

integers

arrays

functions

pointers

The operating system was object oriented

A vector would represent an object

Methods were called by calling one of the functions in the vector

For example, the getcharacter method was in position 0 for a stream

You were required to pass in the vector/object as the first argument of the function

You could write your own stream objects if you followed the conventions

Ethernet

Based on aloha net in Hawaii

All computers on a net on a common wire

Collision detection allowed almost 100% throughput

First transceiver built by outside contractor because PARC had software people and not hardware people

Ethernet protocols

Many protocols tried

PUP - PARC Universal Packet - became the standard for PARC and was used on the ARPANET to some extent

ALTO computers had a hardwired 8-bit address

Initial protocols were designed to work even without routers/gateways or name services

Principle of "best effort" for packet delivery

Routers could trash packets without notification if necessary

User Interface Development

The ALTO was the first convenient personal bitmapped display computer

Many editors, both text and graphical, were tried out

The idea that a document could be edited and seen before it was printed was new

Using many different fonts was new

Bravo editor

single character commands

selection with mouse

user had to be aware of mode command/insertion

Gypsy editor

This was a "modeless" editor - typing always went into the document

The "Laurel editor" was the editor attached to the mail application

instead of "cut" and "paste" the selection point was always used for insertion

typing went in at the insertion point

anything "shift selected" was copied and inserted at the insertion point

anything "control-shift selected" was moved to the insertion point

This allowed pulling many things to the insertion point without losing one's place

Unfortunately, Apple and Microsoft copied Xerox before we got it right

Smalltalk

Based on SIMULA 67 and a remark by Alan Kay, "What if the object oriented paradigm were extended to everything?"

Even integers and booleans were objects in Smalltalk

Smalltalk 72 was an interpreted language because even the parsing of the language could be modified by "methods"

Had a window interface with mouse interaction

(Also a music synthesizer interface)

Smalltalk 76 was compiled into byte codes

Language redefined to allow compilation

interpreted by special microcode in the ALTO

Interlisp-D

Interlisp-D was the extension of interlisp to work with bit mapped displays

Interlisp-D was also byte coded and interpreted by special microcode

Many programming tools were developed for interlisp and it still is comparable to many current software development environments

Interpress and Postscript

Working with the Cedar programming environment John Warnock and Martin Newell find that they have to write a program every time they want to test some graphical code

They develop a simple language, based on forth, to make running experiments easier

They call it JaM for John and Martin

This is seen as a "good thing" and a more structured language, called interpress, is developed to describe printed documents

Xerox, in their infinite wisdom, keeps the language secret for "competitive advantage" resulting in no interpress documents generated by outsiders for Xerox printers and the founding of Adobe by Xerox researchers that don't agree

Postscript is the result at Adobe and is closer to the spirit of the original JaM, i.e., a programming language

People

Chief researcher in early PARC

Butler Lampson

Jack of all trades - helped everywhere

Ed McCreight

Instigator of Smalltalk

Alan Kay

ALTO

Chuck Thacker

EARS

Ron Rider

Interlisp

Dan Bobrow

Larry Masinter