

An ITC Overview

Spring, 1984, I wrote a series of articles for The Tartan outlining the ITC. Copies are attached. (Some refer to an on-going seminar series also held in the Spring.)

Joe Hansen

ITC goal: complete computer command

By W.J. HANSEN

Special for the Tartan

For more than a year the ITC — the Information Technology Center — has been quietly designing a network-of-the-future or CMU. Perhaps it has been too quiet,

since many wonder who we are and what we're up to. This column introduces the ITC and discusses its work.

Physically, the ITC is composed of about thirty people working now in GSIA and Hunt library and is expected to move to the new University Computing Center building early next year. Our goal is to make possible pervasive communication and computing on campus. Every person — student, faculty, and staff — should be able to communicate with any other by a uniform means. Every person should have sufficient computing power to prepare reports, perform computations, and peruse data bases.

To meet this goal we are designing both a

network-based file system and work station software to access that network. Computing has been a vital component of CMU for many years. As President Richard Cyert put it in his charge to the Task Force for the Future of Computing in October, 1981, explosive growth of campus computing has resulted, in part, from "repeated decisions throughout the last quarter century to involve ourselves with this new element — using it in our research, understanding its nature, building it into the fabric of our scholarly lives."

To cope with this explosive growth; CMU signed an agreement with IBM about split fifty-fifty between those previously associated with CMU and new people. Also working for CMU are about a dozen support people, half of them work-study students.

ITC: vice & virtue

By W.J. HANSEN

Special for the Tartan

Campus computing with the ITC system will have better performance and a different appearance, but will have a similarity to the current system: users will execute programs at a workstation and will access the same set of files no matter which workstation they choose.

The basic structure of the system was presented by Jim Morris, director of the Information Technology Center, at a seminar held September 28. Professor Morris did his undergraduate work at Carnegie-Mellon University and currently has an additional appointment in the Computer Science Department.

Morris outlined a system with a shared

network called "Vice" and a workstation for each user. The workstation system is called "Virtue."

Vice is expected to consist of a central "spine" network connecting many "clusters." It is convenient to think of a cluster as being all the workstations on a dormitory corridor or all the workstations in what is now a terminal room. Also attached to the cluster will be a file server and perhaps a printer. The spine will connect all the clusters together with central archival file storage, a gateway to the TOPS systems, central printing, access to other networks, and other amenities.

Virtue workstations will be high performance desktop computers with bitmap dis-

continued on page 8

ITC virtue

continued from page 7

play interface. The operating system will be UNIX (trademark of Bell Laboratories) with a user interface designed by the ITC.

When a user at a workstation operates on a file, the Virtue and Vice systems cooperate to find it. If it is on the user's local disk, that copy is used, otherwise the local cluster server, another cluster server, or the central archive are accessed to get the file. From the user's standpoint, however, the file is requested by name and the system delivers it. It is in this sense that the new system will be much the same as the old.

Items from ITC: The Travels of a File from Vice to You

WJHansen

Special to the Tartan
Information Technology Center, x8880

Why the ITC? Why isn't it enough for everyone to buy their own computer?

A number of colleges have done just that, but CMU needs something more. Everyone needs to be able to *communicate* via mail, files, and other electronic means. This week I want to talk about the ITC approach to communication, a shared file system.

In time-sharing systems - as is the TOPS system now at CMU - each document is stored in a "file" on the computer you are connected to. It is more difficult to access it from some other computer. You may yourself have worked to transfer a file using Kermit, DECnet, or mail; there must be an easier way.

With the ITC's Vice/Virtue system, files are still the key unit of storage, but you will be insulated from concern with their precise physical location.

Suppose you write a report for an English course on your dorm room computer late Wednesday night and have an inspiration Thursday, just before class. You retreat to a workstation in the nearby terminal room and update the report.

How does your file get to the terminal room workstation?

When you finished last night, you stored the report on the network. This means it was sent over the wires in your dormitory to your local "cluster server". When you ask for the file again in the terminal room, the workstation there sends a request to its own local cluster server. The two cluster servers talk to each other and the terminal room workstation gets a copy of the file from the dormitory cluster server.

The process begins when you give the name of the file to the terminal room workstation and it passes it on to its local cluster server. The

name is hierarchical; it has components which refer to increasingly smaller collections of files. An upper element in the name of your report file identifies it as being in your personal collection, so the cluster server looks in its table that describes where the files are for each user. It observes that your files have their home residence on your dorm server and tells the terminal room workstation.

Actual access to the file is done by the workstation. It sends a message over the network to the dorm server and the dorm server sends back the entire file, to be kept on the workstation disk while you work on it. When you are done, the file is returned to its normal residence, your dorm server.

Throughout this process, you have only referred to the file by its name, just as you would with time-sharing today.

July 19, 1984

The time has arrived: 1984 What will happen to us now?

By NEAL W. JOHNSTON
Tartan Features Editor

It's 1984. Does that jog any memories of Big Brother? Or how about the Ministry of Truth? George Orwell, with his insightful novel, *1984*, has made many of us sit back and wonder: could it really happen?

To celebrate the coming of this year, a conference call "George Orwell Days, Pittsburgh 1984" has been created. The idea began with one man: H. Eberhard von Waldow.

"One day about a year ago, I was sitting in the sauna. 'Next year is 1984,'" von Waldow recalled thinking. "I made the connection to Orwell's novel and concluded something needs to be done." So, the conference was conceived, which will examine our world next to Orwell's fictitious totalitarian state.

The conference, a series of workshops, will cover topics that range from the oppression of minorities in totalitarian states and the two faces of technology to the power of mass communication and mind/thought control. The keynote speech, kicking off the conference, will be given by Tom Wicker, associate editor of *The New York Times*.

Von Waldow believes that it is not enough to ask, "Did it come true," or say, "Thank God it did not happen." He views *1984* as more of a warning about what might happen. He believes Orwell's message is: It depends on you.

The conference runs from Friday, March 9 through Sunday, March 11. It will be held at The Pittsburgh Theological Seminary, 6116 North Highland Avenue. For more information call 362-5610. One factor should be noted: all events are free of charge.

MARCH 9-10-11

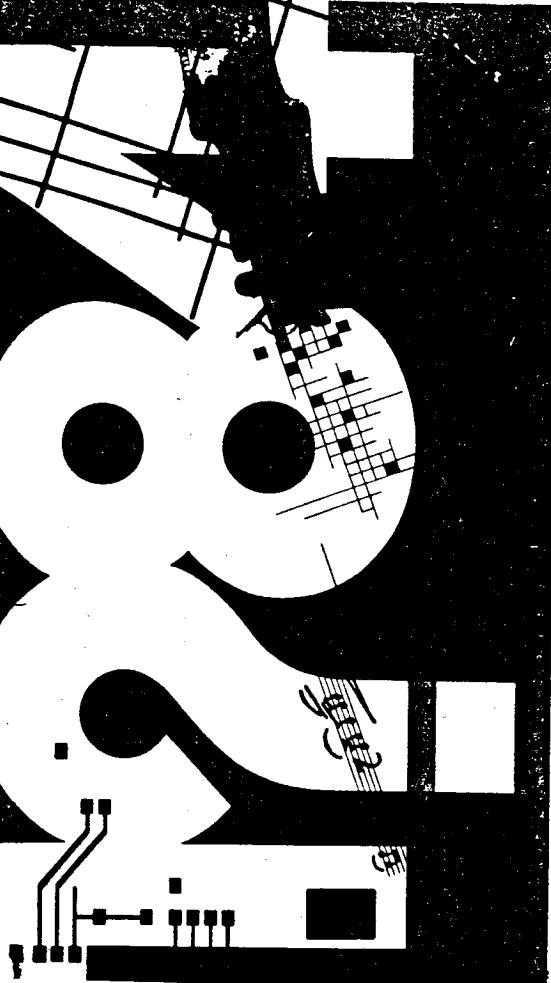


photo by Mike Piewski

Stop and take note: this is 1984.

ITC virtue workstations gain user interface devices

By WILFRED J. HANSEN
Special to the Tartan

In talking about the ITC's Vice/Virtue system, I have mentioned the user interface hardware: keyboard, mouse, and bitmapped display. This week I want to talk a little about what they are and why.

The first thing to say is that these are exactly the same interface devices found on the Macintoshes now selling at a great rate in the campus computer store. So at least one manufacturer and its customers embrace the merits of this interface. Why does the ITC agree?

Keyboard.

Still the best way to get letters, digits, punctuation, and what-not into a

document.

Mouse.

This is a little box about the size of a deck of cards that slides around on the table next to the keyboard. Various mouse designs have one, two, or three buttons on the back of the mouse.

On the screen of the display is an arrow that moves around mimicking your movement of the mouse on the table. When you press one of the buttons, something may happen on the screen at the place where the arrow is pointing: the adjacent character, word or line may be underlined; the image on screen may begin to move; the image may be temporarily replaced with a list of items you can select from; or something else.

In this way you point to pieces of text and do things to them.

Display, Bitmapped.

You talk to computer with keyboard and mouse; it talks back through the display. On older systems the display can typically show 25 lines with 80 characters in each. There are few, if any, ways to show charts, diagrams, and pictures.

In a bitmapped display, the computer program can choose black or white for each individual dot on the screen (sometimes called a pixel). The image space is roughly a thousand pixels wide and almost as high. What does this mean?

With a bitmapped display, a program can display any image that fits, and with resolution

GEORGE
ORWELL
DAYS:
PITTSBURGH
1984

ITC's arrow to mimic the mouse

continued from page 6

significantly changes the possibilities. I mentioned the arrow that mimics the mouse; this is done simply by instructions in the computer that note the mouse position and change the pixels on the screen.

This same speed permits very rapid page flipping as you read a document. There is even the exciting possibility of writing programs to illustrate complex textbook material by showing animated images. Imagine, if you will, an illustration for this report with an image of an arrow moving while you read!

wm: The ITC Window Manager for the Virtue Workstation

WJHansen
Special to the Tartan

Information Technology Center, x6723

James Gosling, developer of the most popular version of EMACS for Unix, is now building '**wm**' and '**be**', as the user interface for the ITC workstation. He talked about this interface at the most recent ITC/CDEC seminar, last week.

To describe '**wm**' and '**be**' let me recall that the user visible parts of the workstation are the keyboard, the mouse, and the mega-pixel bit-mapped display. This screen is far bigger than standard glass teletypes; with the font I am using to write this note, the display screen can show 52 lines of 128 characters each, when none of the screen is taken up with pictures.

With so much screen space, it is nice to be able to have several different items on screen at once. Right now I have this note in the lower left two-thirds of my screen with the system command reader above it. On the right is a list of the files in my current directory and a clock (analog, not digital). The clock is set so it will begin flashing when it is time for me to go home.

'**Wm**' is the traffic cop that divides the screen into areas for the programs I want to be running. Each area is called a window; thus '**wm**' is short for 'window manager'.

In addition to splitting the screen, **wm** assists programs written for the system. It displays characters in any of (right now) some fifty fonts and draws lines between any two points. The clock face, for example is made of 21 lines and 48 dots. Another service **wm** provides is to decide which program gets keystrokes from the keyboard and button pushes from the mouse.

The mouse is generally used to point to items on the screen. I can point to the start and end of a phrase and then erase it with one keystroke. By pointing at a line between windows, I can "pick it up" and move it to make the window on one side bigger and the other smaller.

When I press the middle button on the mouse, a rectangle temporarily replaces the part of the screen I am pointing at. It contains a 'menu', a list of operations I can ask to occur. For example, when using 'be' I can ask to cut the currently selected phrase, paste in the most recently cut item, write the file, exit, or even change the appearance of the current phrase: bold, italic, bigger, indented,

So what is 'be'? It is an 'e'ditor designed to serve as a 'b'asis for applications; thus true text editor in that it moves words from line to line so each line is as full as it can be. As a basis for applications, its most important property is that text cut from one window can be pasted in another.

I will write another week about the many wonders of the base editor and its multiple applications.

March 9, 1984

Viewing Your Work through Base Editor Colored Glasses

WJHansen, Information Technology Center, x6723

Special to the Tartan

One of life's special frustrations is losing your house key. You usually don't find out until it is time for supper and you are already too hungry to cope rationally. Using a new computer system can have a similar frustration: you know the system has almost boundless power, if only you had the keys to open it.

The frustration is compounded on 'old-style' systems when each program you run opens to a different key. The design of the user interface for the ITC's workstation software is intended to solve this problem; many applications will interface to you through the agency of 'be', a *base editor*, whose keys are designed for ease of learning and use.

reading sections, searching for items, copying pieces, adjusting the formatting, and making changes. Before describing the interaction of 'be' and applications, let me describe how I used old-style, pre-be, systems for three tasks: system commands, HELP, and file reorganization:

In old-style systems, the only way to make corrections as you enter system commands is to use backspace. If you have hit RETURN, you probably have to retype the entire line. Moreover, you can only see previous work as far back as the top of the screen,

The HELP command often just displays the contents of a file on the screen. If the file is bigger than the screen, you have to start over to go back to any previous portion of the file.

If you are like me you reorganize your files occasionally: go through them, delete some, and move others. To do this with old-style systems, I had to list the names of files and give some Delete or Rename commands. After a few such, however, the file names moved off the top of the screen and I had to start over again.

Nowadays, I am using a prototype of the ITC Virtue system. I manage each of these three tasks through the agency of 'be'. I give system commands in one window, view help in another, and reorganize files in a third. Yet all three windows are controlled by a single editor that provides commands for moving around the image, inserting and deleting pieces, and other editing operations.

By pointing at entries in the file list, I can delete or rename files and still have the list. I can also point at an entry and ask to see its contents; I never have to give a command to ask to edit a file.

When I ask for help or give system commands the base editor lets me move through the help file or commands as easily as I can move through any document. Indeed, I use exactly the same commands. When I see an example in a help document, I can cut it out and give it as a command to the system. When I want to give a command again, I can copy it from the recorded history of system commands and even make changes before doing it again.

Even if the base editor were not providing tremendous advantages to users of applications that go through it, it still is a great advance over earlier editors. Essentially it provides the most common Scribe(TM) formatting features dynamically. I can look at my text and see it italic, bold face, indented, centered, enlarged.... When displaying a paragraph, the lines are automatically adjusted so each is flush at both left and right. A plain ASCII file is a document for the base editor, but so is a file containing font changes, line drawings, tables, and graphic images.

Let us turn to another application of the base editor: mail. When I want to send a message to, say, 'Loche Smith @ CMU', I call 'Send', which displays a window with a blank header and space for the message text. I either type the address or copy it over from somewhere in another window. I can either type the text or copy parts of it from another window. To emphasize sections, I can even make them bold: "Please hurry over with keys."

When I want to read my mail or bulletin boards, I call on 'Scan'. It displays one window with the headers of messages and another with the text of one of those messages. By pointing with the mouse I can select any header and that message will be displayed.

As the system evolves, numerous other applications will utilize the 'be' interface. This will make it easier to write these applications because they need not write their own interaction routines. More important, each user will know immediately the keys to open that application.