CPE/CSC 486: Human-Computer Interaction

Franz J. Kurfess

Computer Science Department
California Polytechnic State University
San Luis Obispo, CA, U.S.A.

Course Overview

- Introduction
- Cognitive Foundations
- Input-Output Devices
- Interaction Spaces
- Interaction Styles
- Interaction with Mobile Devices
- Speech-Based Interaction
- User Assistance
- Natural User Interfaces
- Case Studies
- Project Presentations

Chapter Overview Interaction Styles

- Motivation
- Objectives
- Terminology
- Batch Systems
- Command-Line Interfaces
- Full-Screen Interfaces
- Menus
- Forms
- Direct Manipulation Interfaces
- WIMPs
- User Assistance
- Interaction through Speech
- Interaction with Mobile Devices

© Franz J. Kurfess

Interaction Styles in CSC 484

- **there is a similar chapter in the CPE/CSC 484 class**
  - follows the Rogers, Preece, Sharp textbook
  - see http://users.csc.calpoly.edu/~fkurfess/Courses/484/S12/Administration/Schedule.html, Week 3

Logistics

- **Use of HCI Lab Facilities**
  - Morae
  - reservations for exclusive use of the lab
- **Open House: Fri, April 13 + Sat, April 14**
  - opportunity for usability evaluations and data collection
    - Fri ~2:30 - 4:00 pm: new students, parents
- **Loaner Devices Checkout**
  - iPads, XBox + Kinect, PS3 + Move, Qualcomm Android kits
- **Assignments**
  - A1 due today
    - project-related => TRAC Wiki
    - others => PolyLearn Assignment Submission
  - A2 published (same as 484 - A4)
- **Research Activity**
  - topic selected?
  - dissemination method discussed (paper, blog, video)
- **Term Project**
  - addition of students who enrolled late
  - contact and regular meetings with external customers

© Franz J. Kurfess

Motivation

- the way interaction with computers is performed depends on available technology on one hand, but also on different methods and usage styles for a particular technology

- the change in interaction style from command-line to graphical user interface has contributed considerably to the popularity of personal computers

- for a given task and user population, different interaction styles may make the difference between success, acceptance, or failure of a product

© Franz J. Kurfess

Objectives

- to know the advantages and drawbacks of the most often used interaction styles, in particular command-based vs. graphical user interfaces
- to be familiar with evaluation criteria for a comparison of different interaction styles
- to be able to select an appropriate interaction style for a specific task, environment, and user population
- to be exposed to emerging interaction styles like natural language, gestures, or intelligent agents

Interaction

- exchange of information between user and system
- actions of the user that change the status of the system
- feedback to the user concerning actions of the system
- requires translation between the intentions of the user and the actions of the system

Terminology

- **task**
  - actions to be performed in order to solve a problem in an application domain
- **goal**
  - desired output from a completed task
- **task analysis**
  - identification of the problems space
    - in particular domain, goals, intentions, specific tasks
- **user language (task language)**
  - describes the problem to be solved in terms familiar to the user
- **system language (core language)**
  - describes the functionality of the system in terms familiar to the designer or developer

© Franz J. Kurfess

Interaction Model

- **execution**
  - establishing the goal
  - forming the intention
  - specifying the action sequence
  - executing the action
- **evaluation**
  - perceiving the system state
  - interpreting the system state
  - evaluation of the system state
    - with respect to goals and intentions

[Norman, Dix]

Interaction Styles

- **term covers all of the ways that users interact with a computer system**
  - also referred to as communication styles or dialog styles
- **represent alternative design strategies for the UI**
  - each style offers its own way of organizing the system’s functionality, of managing the user’s inputs, and of presenting information
    - e.g. display-based interfaces -> menus, mice, windows, widgets, icons, buttons, function keys, etc.
- **provide a behavioral view of how the user communicates with the system**
  - “look and feel”

[Mustillo]

Evaluation Criteria for Interaction Styles

- **representational**
  - how a system “looks and feels” (graphical), or “sounds and feels” (speech)
- **operational**
  - how a system links sequences of operations together
  - maps the representations used onto its functions
- **navigational**
  - awareness of what users can do at each stage
  - how they can move through the system

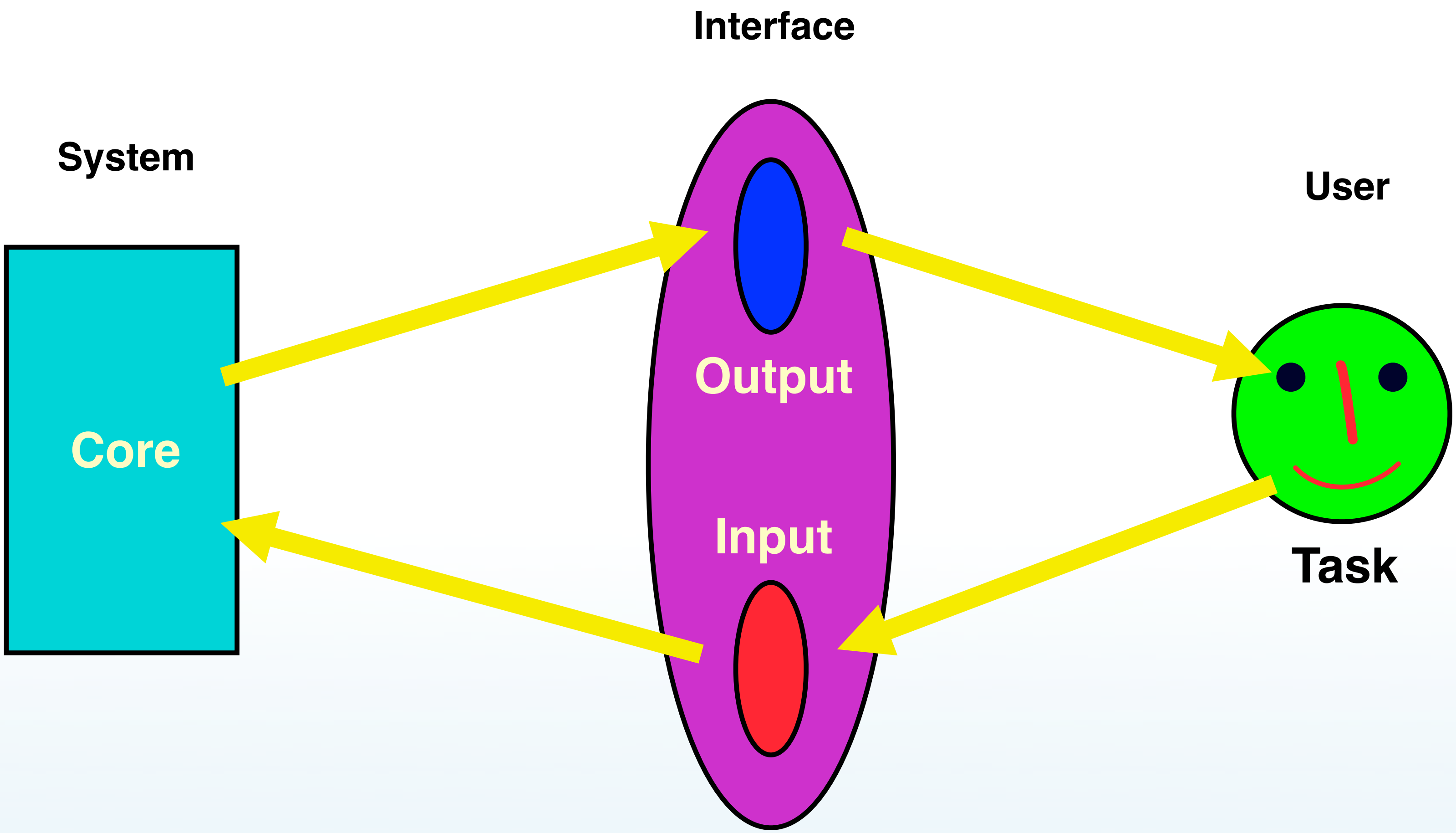
[Mustillo]

Interaction Framework

- **components**
  - system user, input, output
  - input and output together form the interface
  - each of the components may have its own language to describe the objects and actions it is concerned with

[Dix]

Interaction Framework



Interface

System

User

Core

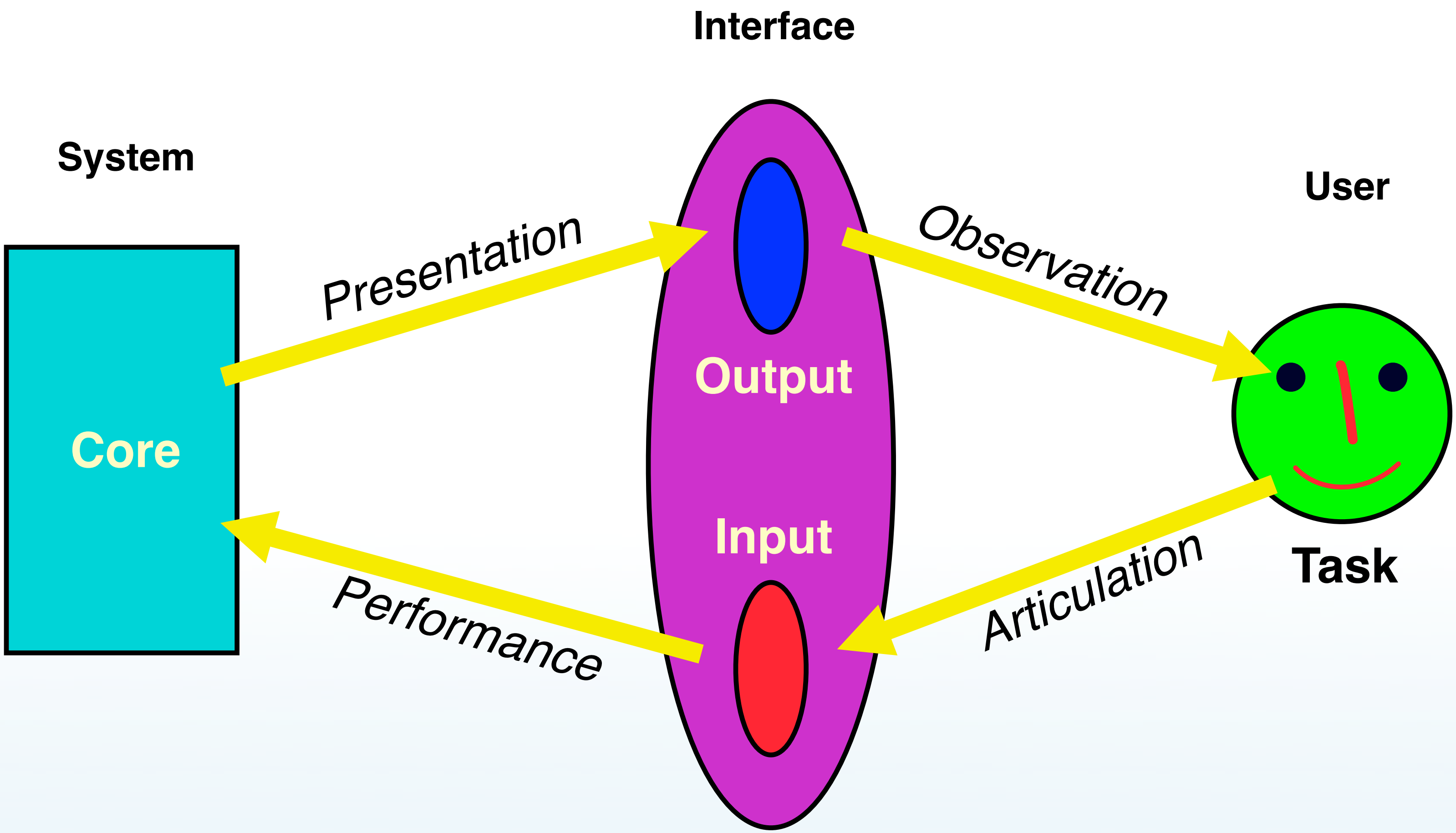
Output

Input

Task

[Dix]

Interactive Cycle



[Dix]

Translations Between Components

- **articulation**
  - user translates task intentions into the input language
- **performance**
  - input language is translated into stimuli for the system
- **presentation**
  - system activities are translated into output language
- **observation**
  - output language is translated into the user’s task model

Example: Light in a Room

- **controlling the lighting in a room**
  - articulation: “I’m going to bed now, so I better turn off the light in the living room. To do this, I need to flip the switch.”
  - task language: turn lights on/off
  - input language: flip switch
  - system language: close/open circuit for light bulbs
  - output language: lights on/off

Example (cont)

- **translations**
  - articulation
    - user decides to turn on the light, and flips a switch
  - performance
    - flipped switch closes the circuit
  - presentation
    - light bulb emits light
  - observation
    - user notices that the light is on
- **frequent problem**
  - multiple switches in large rooms

Practical Exercise: Digital vs. Analog Clocks

- **Describe and compare the languages and translations used for setting digital and analog clocks. What are frequent problems with the two device types and their languages or translations?**
- **languages**
  - task language: set time to a certain value
  - input language: wheel vs. buttons
  - system language: mechanical movement vs. ICs
  - output language: minute and hour hands vs. LCD display

© Franz J. Kurfess

Practical Exercise (cont.)

- **translations**
  - articulation
    - turn wheel / press buttons
  - performance
    - clock translates input actions into modified display
  - presentation
    - new time setting is displayed
  - observation
    - user translates hand positions or numbers into time

Exercise: Select Example

- **Identify an activity or scenario to illustrate interaction styles**
  - languages
    - task, input, system, output
  - translations
    - articulation, performance, presentation, observation
  - problems

Practical Exercise (cont.)

- **problems**
  - which way do you turn the wheel
  - which button do you press, how often, in which combination
  - turning the time “back” on a digital clock is frequently impossible

Batch Systems

- first generation of user interfaces
- interaction restricted to a single point in time
  - submission of a batch job as a single unit
- user commands have to be specified before the result of any of them is made known to the user
- work well for repetitive jobs (e.g., payroll processing, billing, etc.), and are still used today
  - Examples: Revenue Canada, Canada Post, Bell Canada.
- drawbacks
  - not interactive
  - inflexible

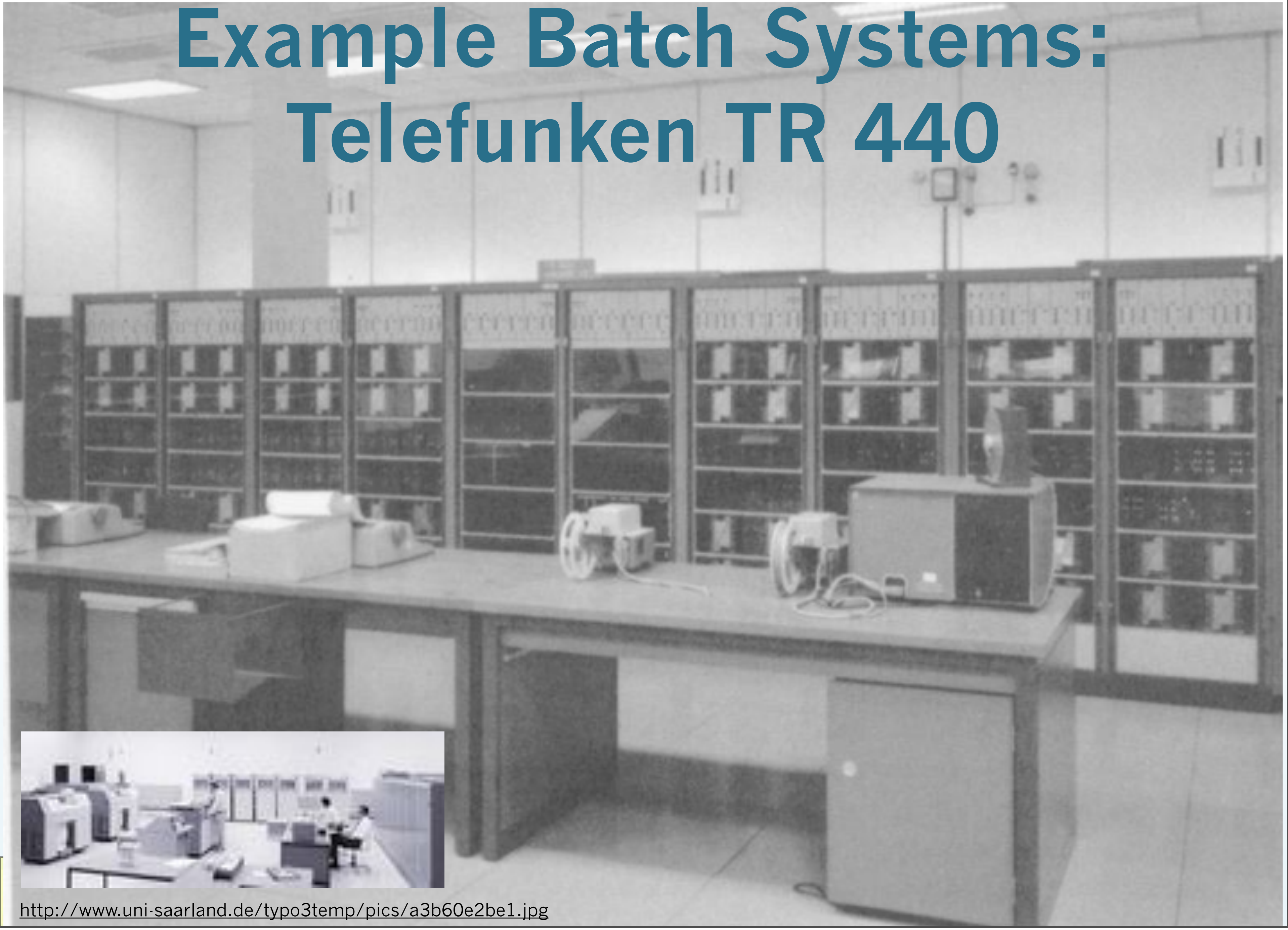
CAL POLY

[Mustillo]

Personal History

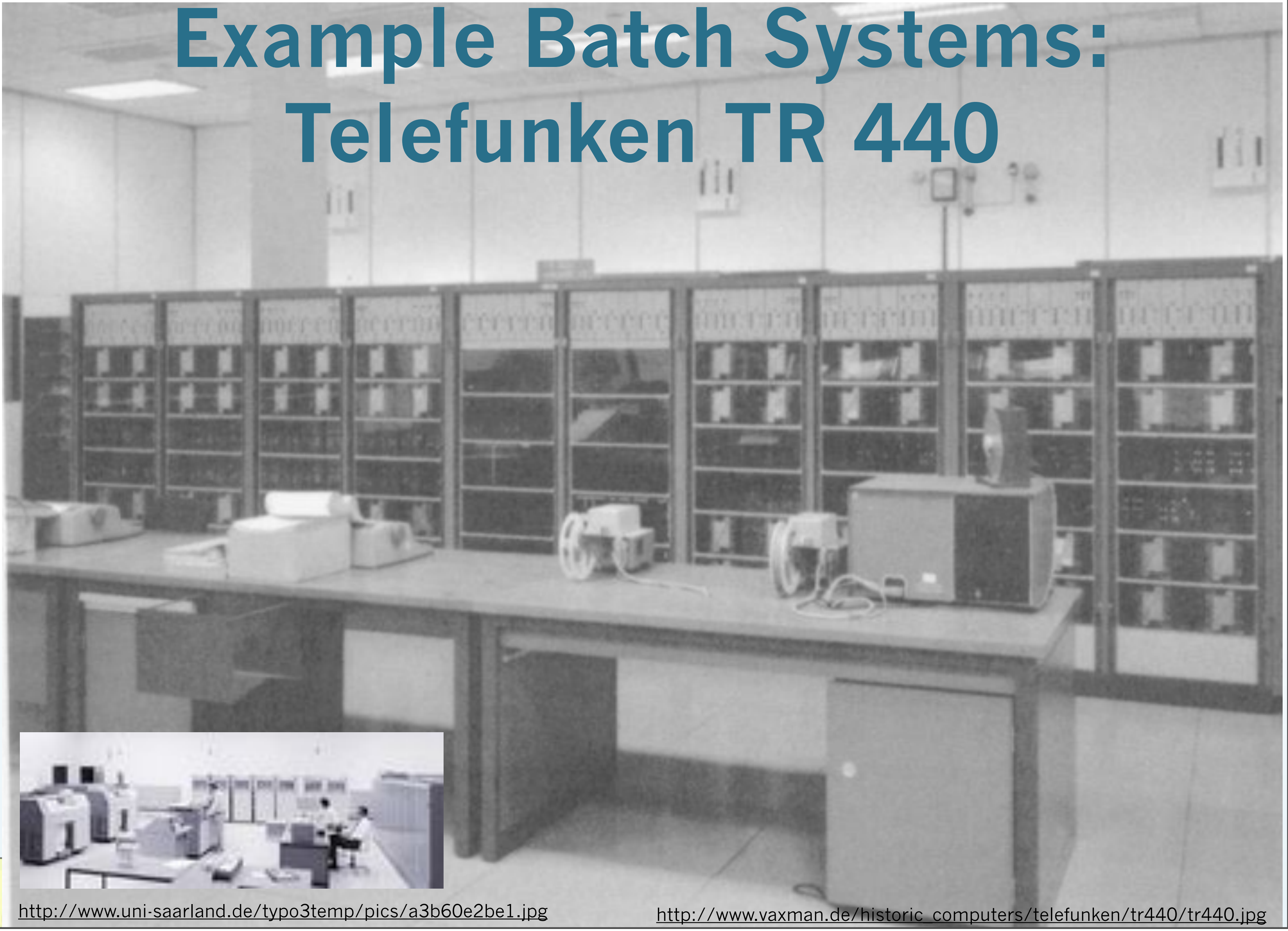
- the images in the following slides represent computer systems I used early in my career

Example Batch Systems: Telefunken TR 440



http://www.uni-saarland.de/typo3temp/pics/a3b60e2be1.jpg

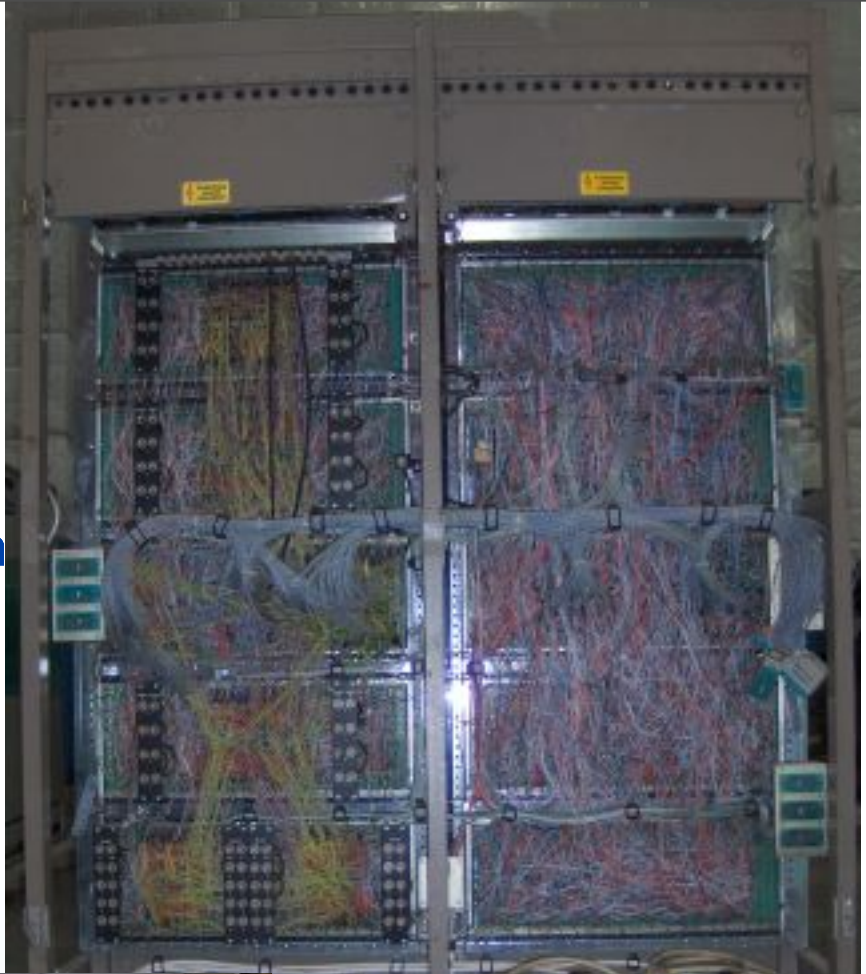
Example Batch Systems: Telefunken TR 440



http://www.uni-saarland.de/typo3temp/pics/a3b60e2be1.jpg

http://www.vaxman.de/historic_computers/telefunken/tr440/tr440.jpg

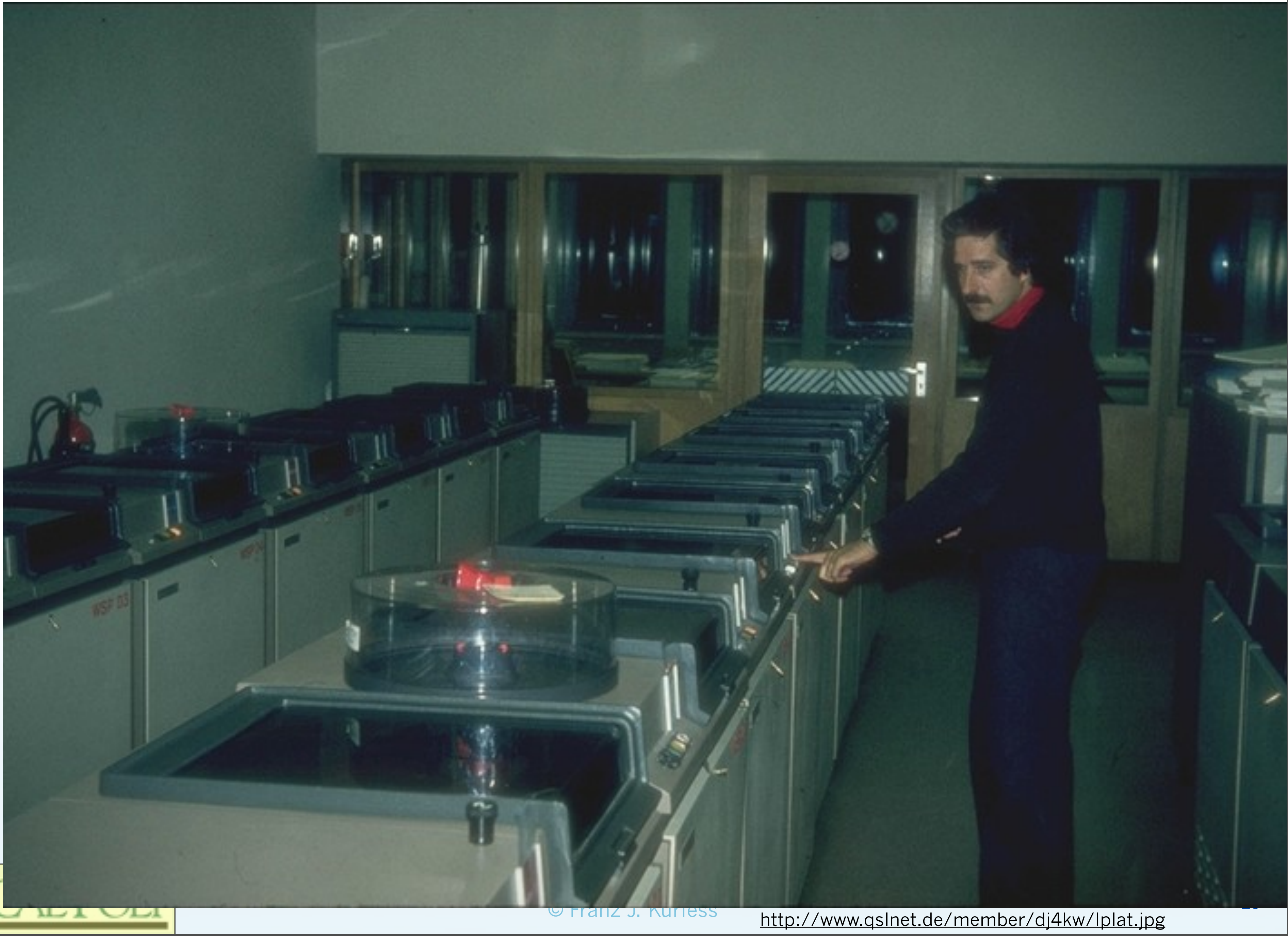
- **TR440**
  - mainframe,
  - communication cabinet
  - console



http://archive.computerhistory.org/resources/physical-object/telefunken/102667999.5.lg.jpg?rand=668344533

http://archive.computerhistory.org/resources/physical-object/telefunken/102668025.1.sm.jpg

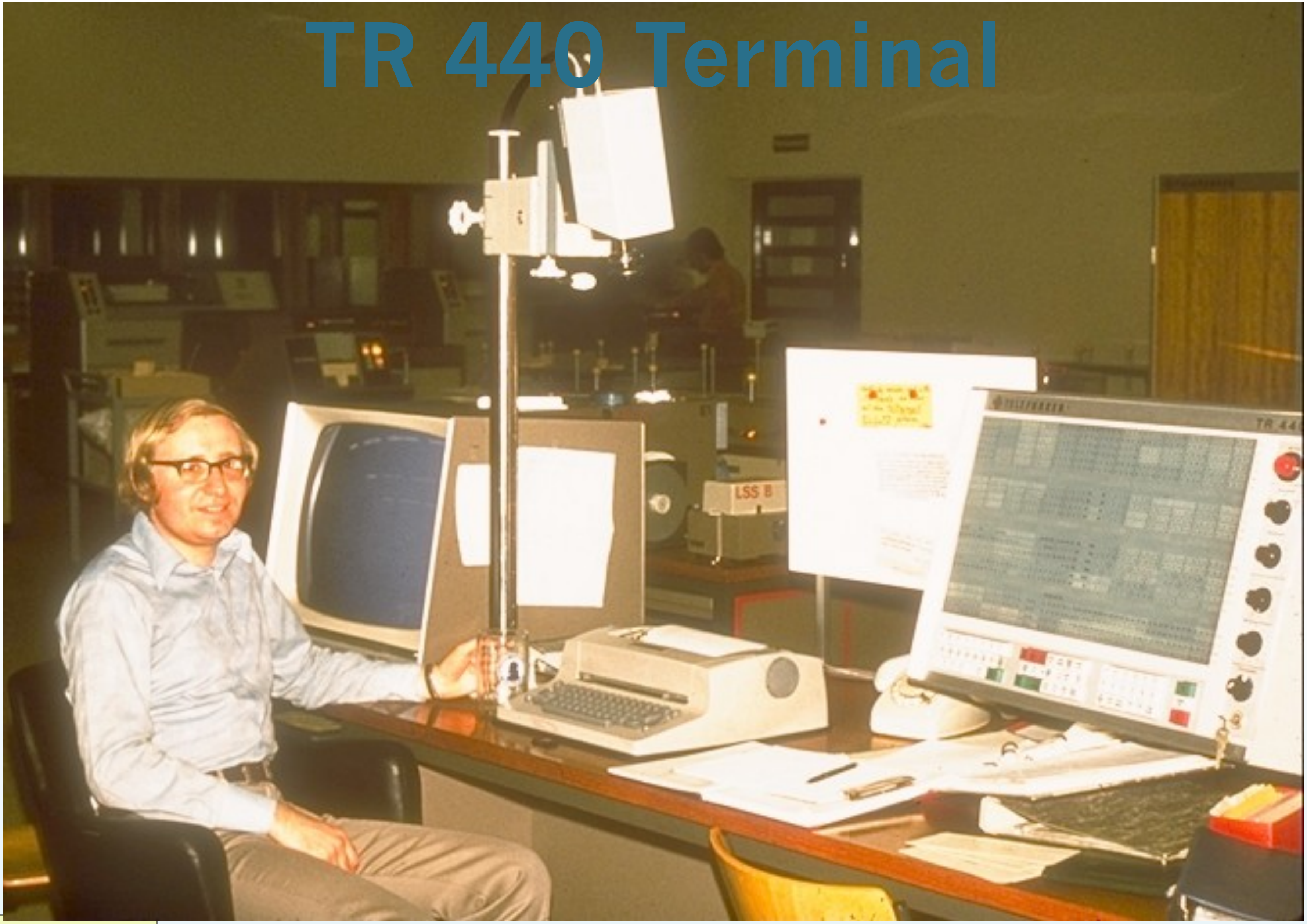
http://www.qslnet.de/member/dj4kw/lt440.jpg



© Franz J. Kurfess

http://www.qslnet.de/member/dj4kw/lplat.jpg

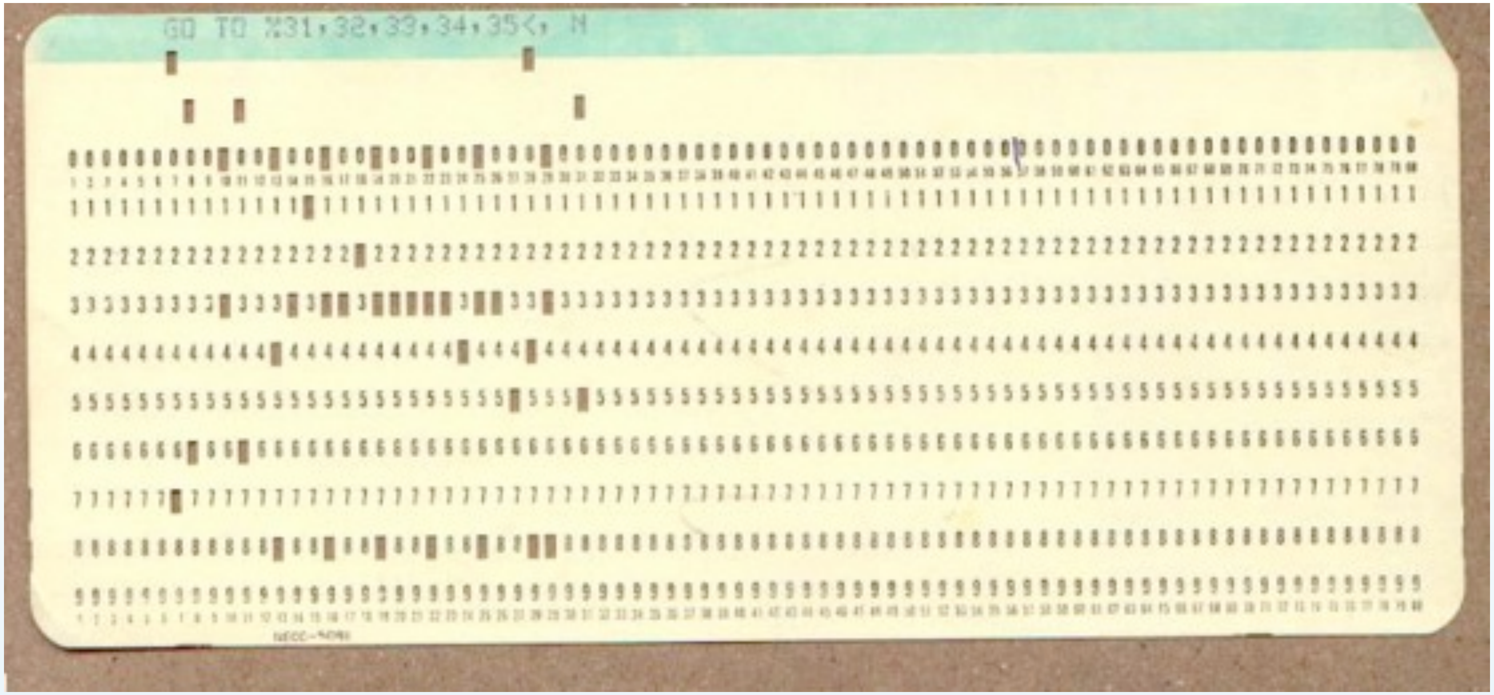
TR 440 Terminal



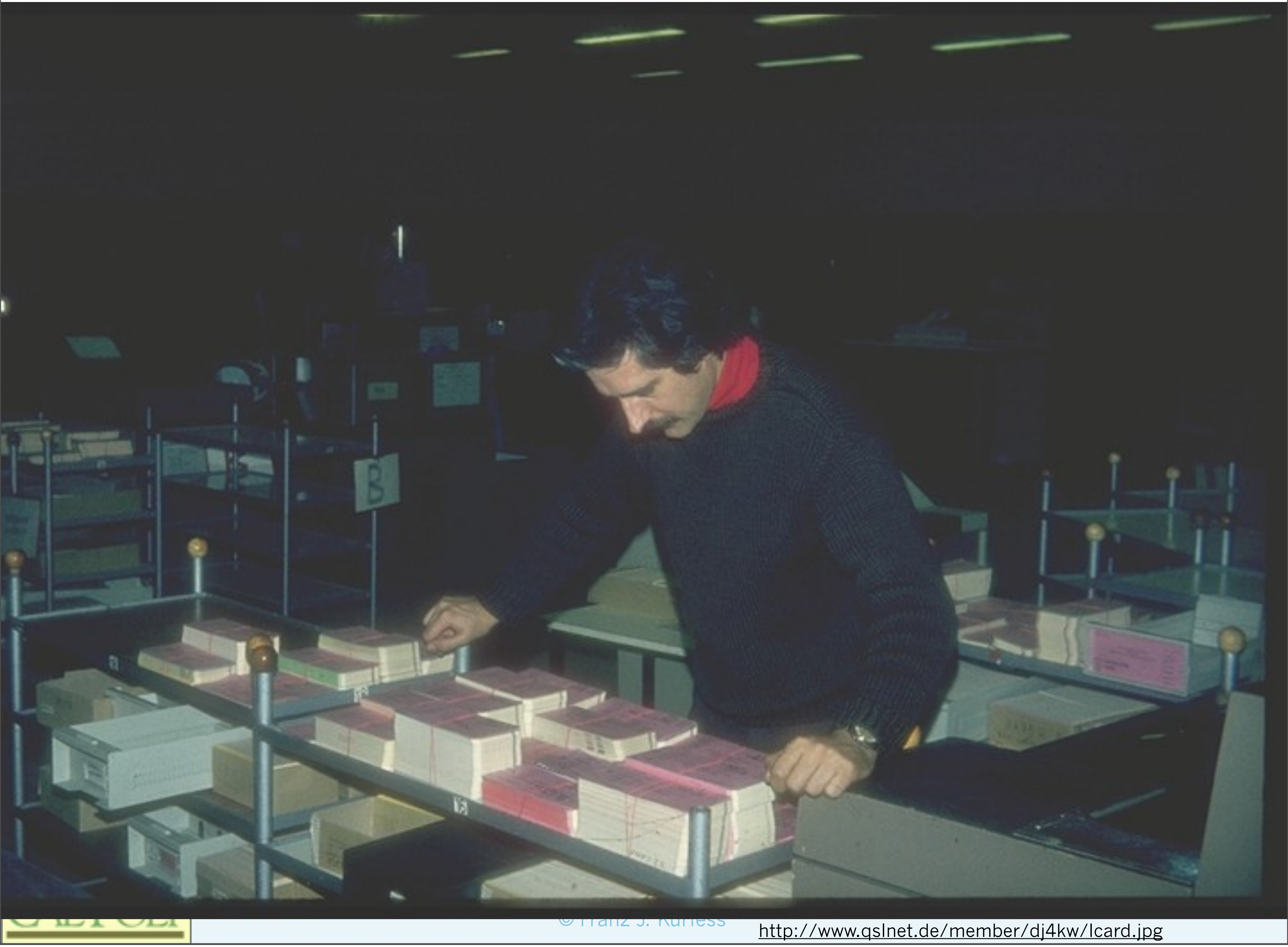
http://www.qslnet.de/member/dj4kw/ljim.jpg

Example Batch Systems

Example Batch Systems



http://www.epa.gov/athens/learn2model/part-one/career/images/card3.jpg



© Franz J. Kurfess

http://www.qslnet.de/member/dj4kw/lcard.jpg

http://www.qslnet.de/member/dj4kw/loink.jpg

Command-Line Interfaces

- **also referred to as Command-Language or Line-Oriented Interfaces**
- **A new user's first view of almost all command-line interfaces: “>“**
- **one-dimensional**
  - user interacts with a system on a single line that serves as the command line
  - once the user hits the return key, the input can no longer be modified
- **typical of many early computer systems, but remain common even today**
  - e.g., MS-DOS, UNIX

[Mustillo]

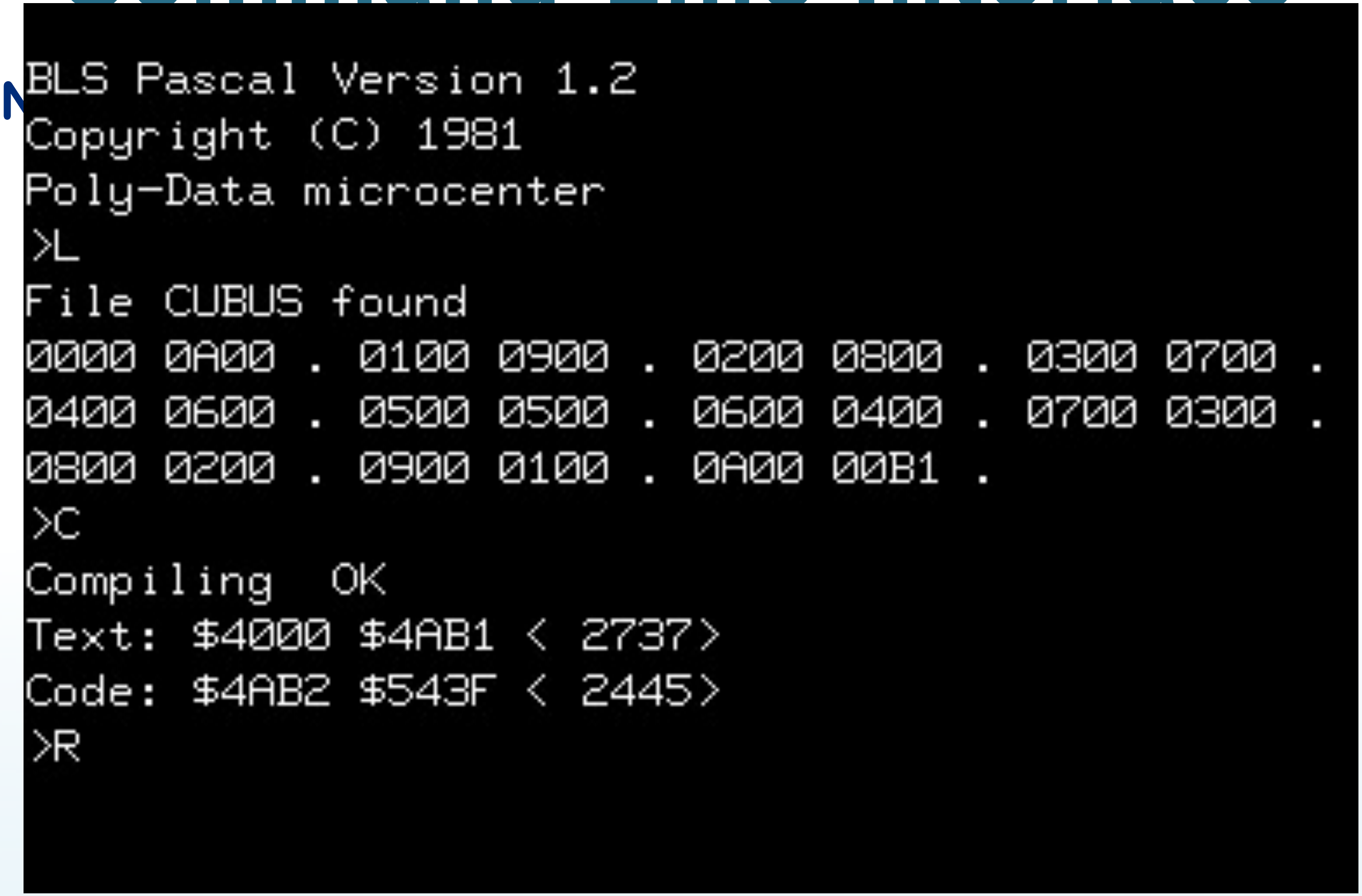
Command-Line Interfaces (cont.)

- **responsibility for navigation is on the user**
  - user has to know what the allowable commands are
  - needs to have a clear idea of the function to be performed
- **difficult for novice users to learn**
  - once mastered, command languages often represent the quickest form of communication
    - abbreviations, function keys, keyboard shortcuts,
- **query languages represent a special case of command languages**
  - allow users to request information

[Mustillo]

Example 1
Command-Line Interface

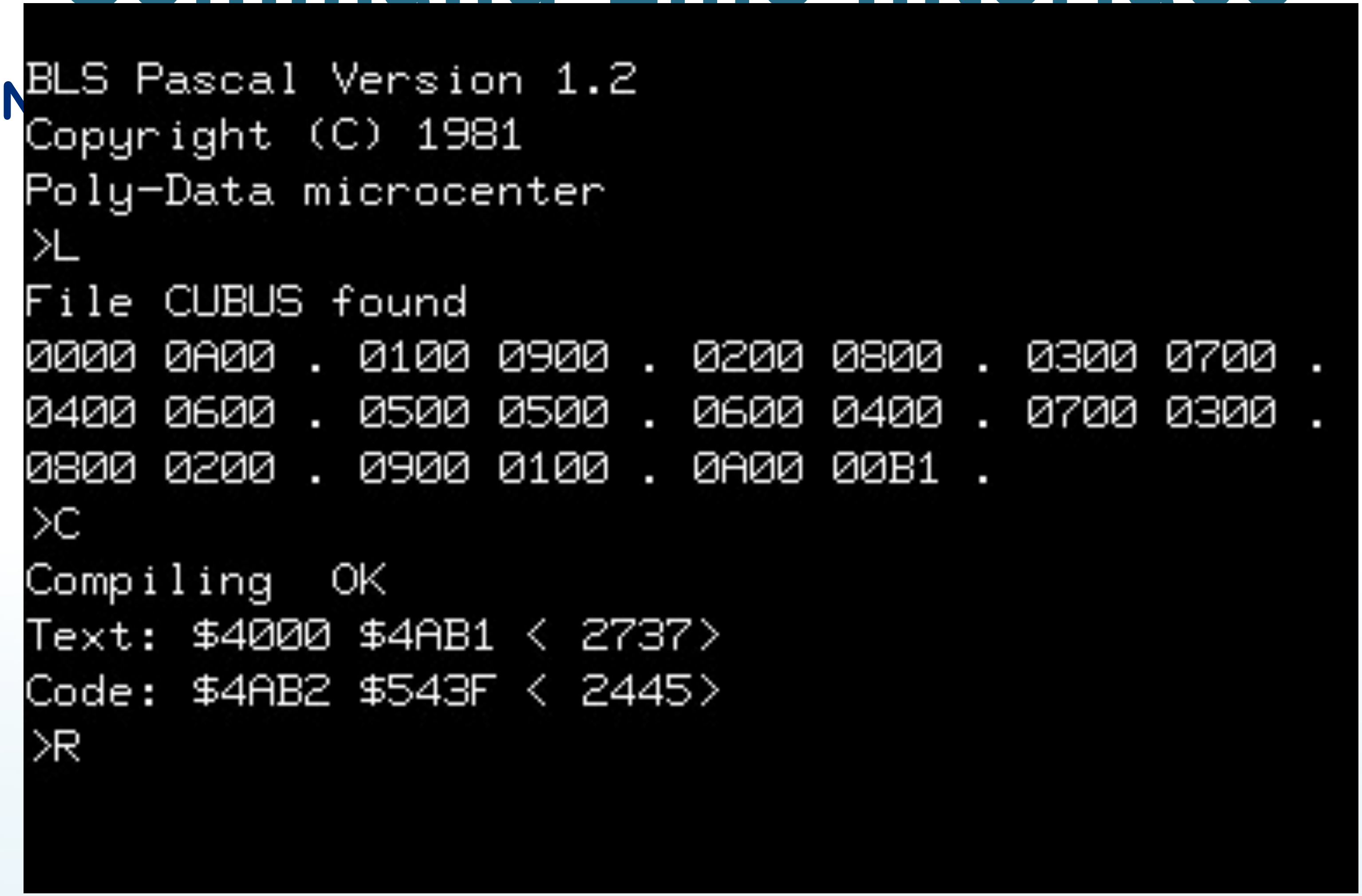
- N



```
BLS Pascal Version 1.2
Copyright (C) 1981
Poly-Data microcenter
>L
File CUBUS found
0000 0A00 . 0100 0900 . 0200 0800 . 0300 0700 .
0400 0600 . 0500 0500 . 0600 0400 . 0700 0300 .
0800 0200 . 0900 0100 . 0A00 00B1 .
>C
Compiling  OK
Text: $4000 $4AB1 < 2737>
Code: $4AB2 $543F < 2445>
>R
```

Example 1
Command-Line Interface

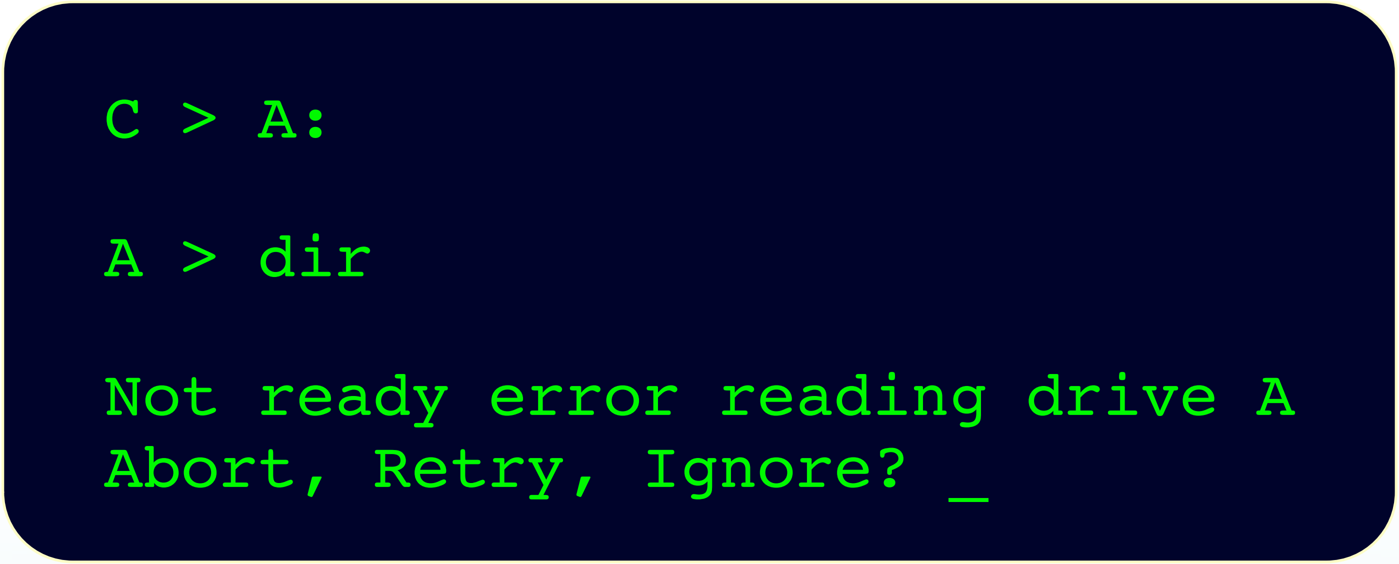
- N



```
BLS Pascal Version 1.2
Copyright (C) 1981
Poly-Data microcenter
>L
File CUBUS found
0000 0A00 . 0100 0900 . 0200 0800 . 0300 0700 .
0400 0600 . 0500 0500 . 0600 0400 . 0700 0300 .
0800 0200 . 0900 0100 . 0A00 00B1 .
>C
Compiling  OK
Text: $4000 $4AB1 < 2737>
Code: $4AB2 $543F < 2445>
>R
```

http://www.myplace.nu/nascom/images/simfull.jpg

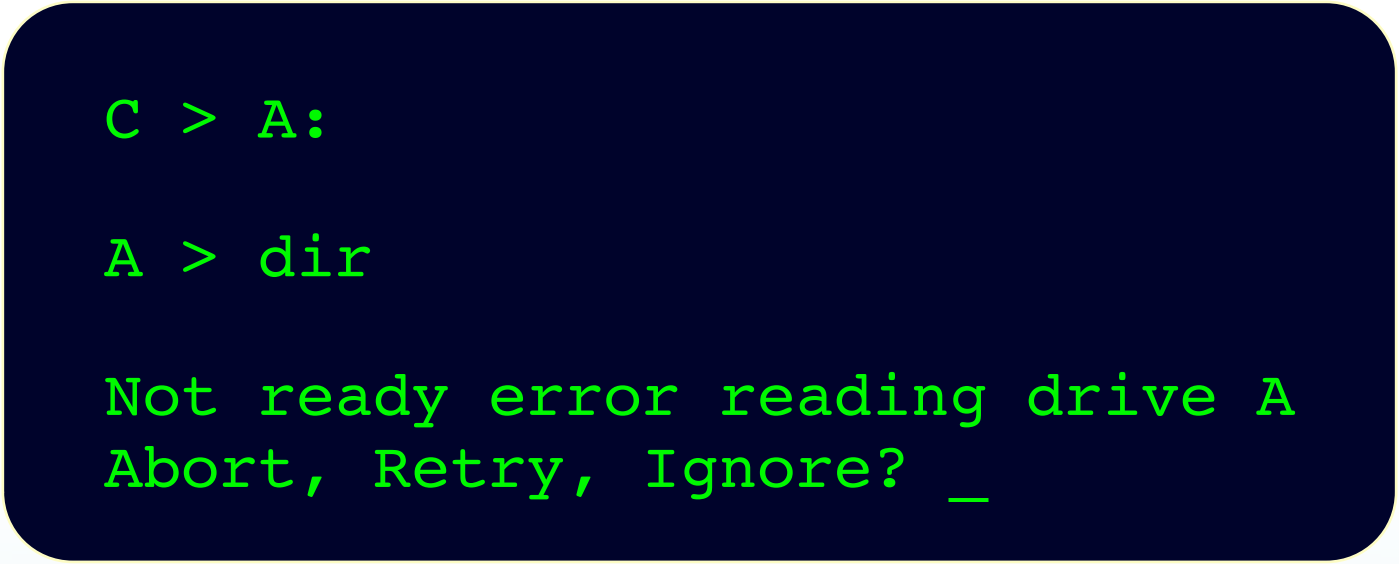
Example 2
Command-Line Interface



```
C > A:

A > dir

Not ready error reading drive A
Abort, Retry, Ignore? _
```

[Mustillo]

Example 2
Command-Line Interface

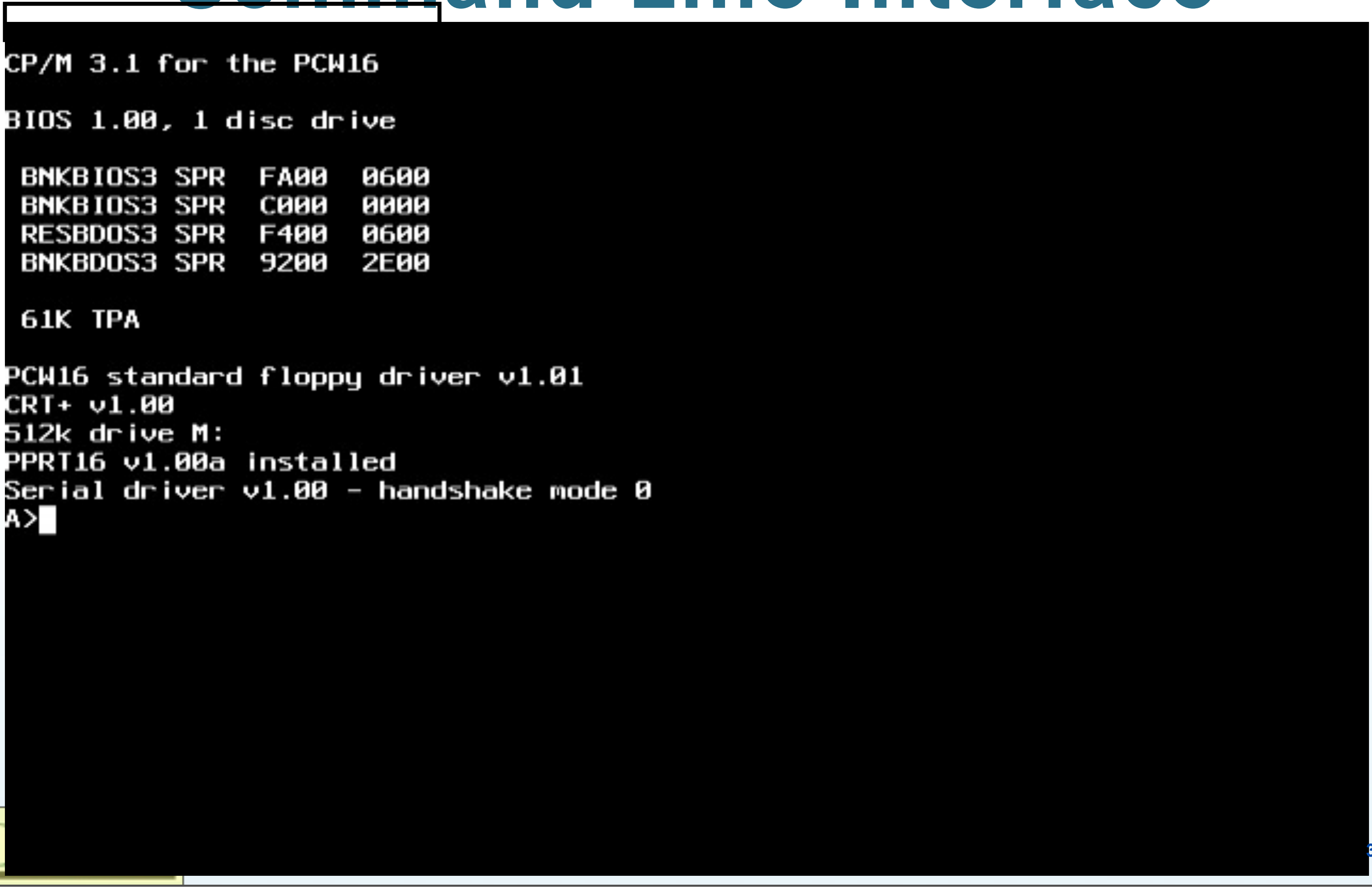


```
C > A:

A > dir

Not ready error reading drive A
Abort, Retry, Ignore? _
```

A typical dialog using an MS-DOS system

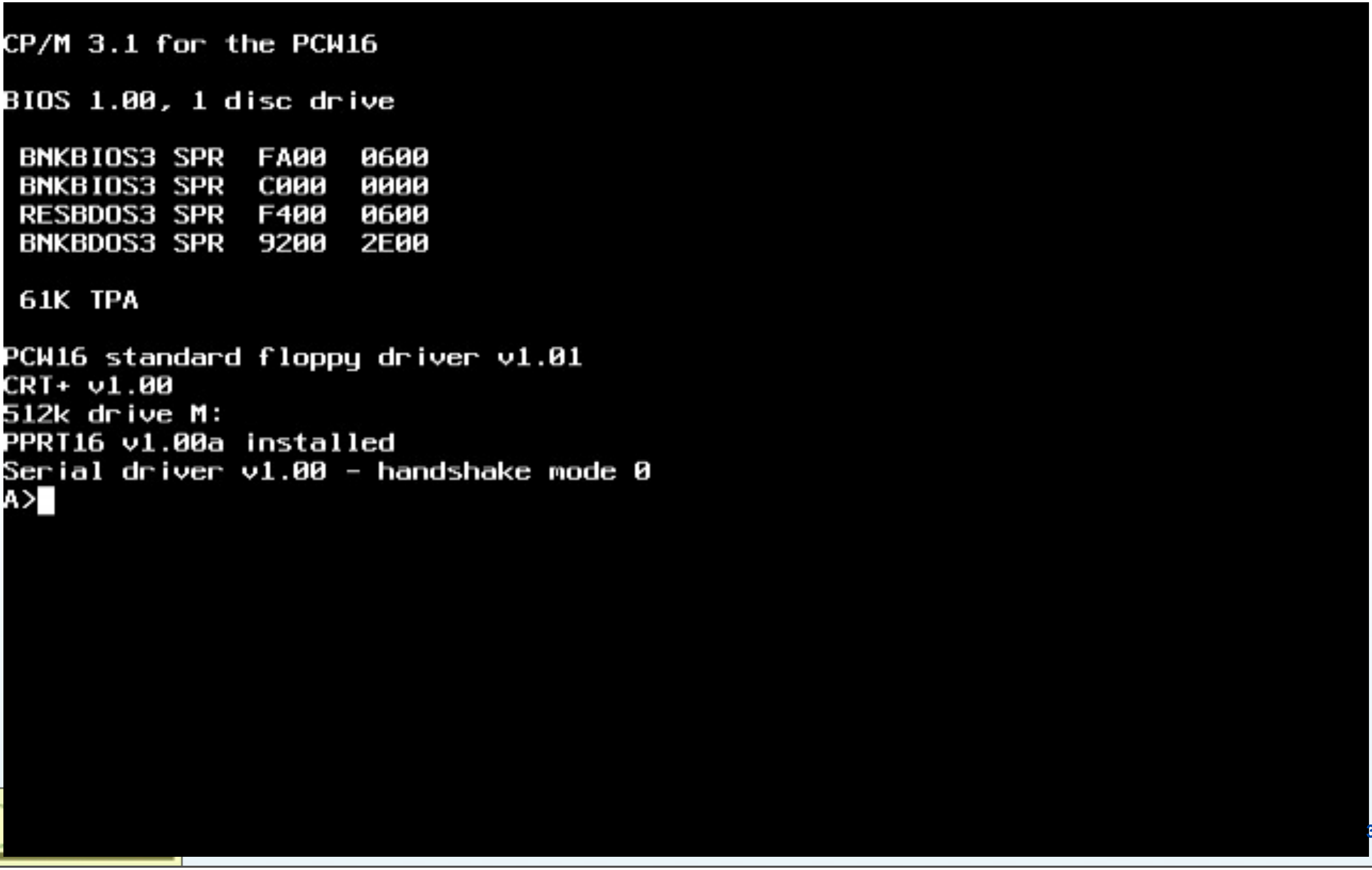
[Mustillo]

Example 3
Command Line Interface



```
CP/M 3.1 for the PCW16

BIOS 1.00, 1 disc drive

 BNKBIOS3 SPR   FA00   0600
 BNKBIOS3 SPR   C000   0000
 RESBDOS3 SPR   F400   0600
 BNKBDOS3 SPR   9200   2E00

 61K  TPA

PCW16 standard floppy driver v1.01
CRT+ v1.00
512k drive M:
PPRT16 v1.00a installed
Serial driver v1.00 - handshake mode 0
A>
```

Example 3
Command Line Interface



```
CP/M 3.1 for the PCW16

BIOS 1.00, 1 disc drive

 BNKBIOS3 SPR  FA00  0600
 BNKBIOS3 SPR  C000  0000
 RESBDOS3 SPR  F400  0600
 BNKBDOS3 SPR  9200  2E00

 61K TPA

PCW16 standard floppy driver v1.01
CRT+ v1.00
512k drive M:
PPRT16 v1.00a installed
Serial driver v1.00 - handshake mode 0
A>
```

Example 4
Command-Line Interface

```
HP-UX bmtlh730 A.09.05 A 9000/755 (ttyw0)

login: fkurfess
Password: **********
Please wait...checking for disk quotas

mtlh730:~> cokol
Enter EDMX password:

mtlh730:~> exit
```

[Mustillo]

© Franz J. Kurfess

Example 4
Command-Line Interface

```
HP-UX bmtlh730 A.09.05 A 9000/755 (ttyw0)

login: fkurfess
Password: **********
Please wait...checking for disk quotas

mtlh730:~> cokol
Enter EDMX password:

mtlh730:~> exit
```

A typical dialog using a UNIX system (Telnet session)

[Mustillo]

© Franz J. Kurfess

Advantages Command-Line Interfaces

- **powerful**
  - commands can be stringed together, automated scripts can be run, macros can be used, commands can be executed repetitively)
- **flexible, user controlled**
  - users can do anything in any order
- **fast, efficient**
  - requires more typing than other interaction styles, but is probably the fastest, most efficient of all interaction styles
  - very fast and efficient for expert, high-frequency users)
- **use minimal screen space**
  - only one line at a time

[Mustillo]

Drawbacks Command-Line Interfaces

- **operating system or program exposed to the user**
- **difficult to learn**
  - very cryptic, use of arbitrary syntactic delimiters, require rote memorization
- **difficult to remember**
  - especially for infrequent users
  - rely totally on recall
- **inflexible**
  - commands need to be expressed using a precise syntax
  - any variation represents an error
- **susceptible to errors, poor error handling**

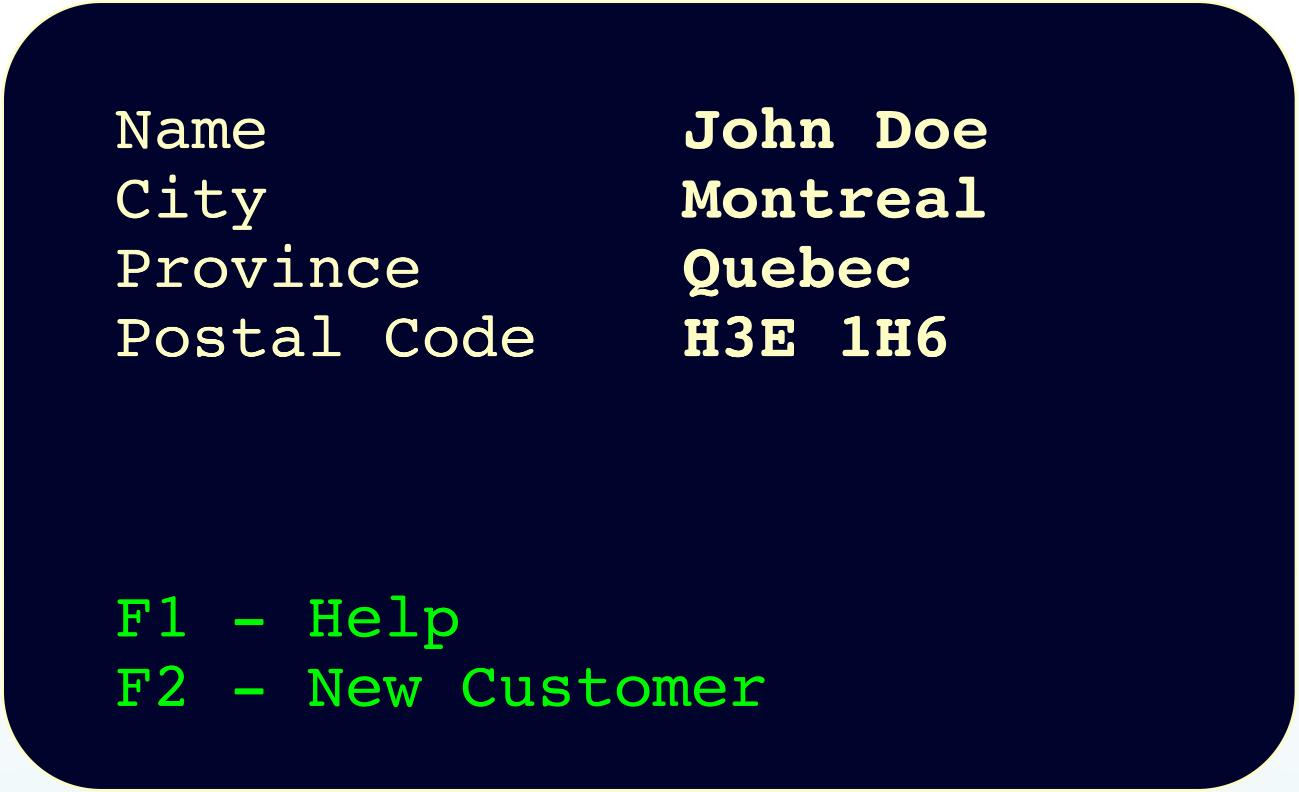
[Mustillo]

Full-Screen Interfaces

- **two-dimensional**
  - user is no longer limited to a single command line, but can move around the screen along two dimensions
    - e.g. form-filling application, in which the user is presented with a number of labeled fields that can be edited in any sequence.
- **many full-screen interfaces use function keys, enumerated choices, or the first characters of a menu item as the primary interaction style**

[Mustillo]

Example Full-Screen Interfaces



```
Name            John Doe
City            Montreal
Province        Quebec
Postal Code     H3E 1H6



F1 - Help
F2 - New Customer
```

Example of a full-screen interface

[Mustillo]

Menus

- mechanism that allows the user to make a selection from a limited set of options
- one of the most popular interaction styles
- the user does not have to remember the name or abbreviation of a command
  - recognize it from a list of available options
- names of options, contents of the icons, descriptions of buttons should be self-explanatory

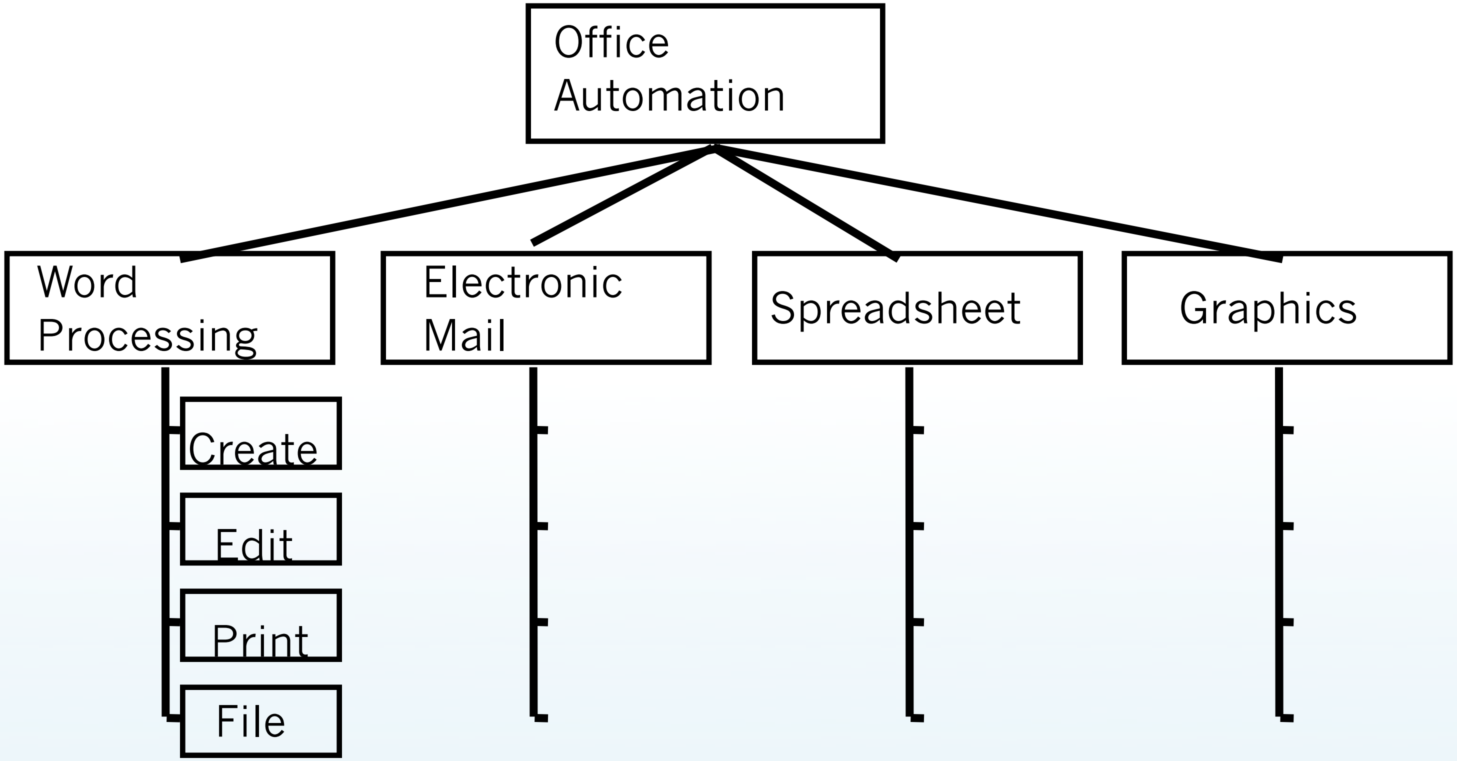
[Mustillo]

Menu Building Blocks

- **menus can consist of**
  - textual descriptions of the available functions
  - icons
    - small graphic images that represent different aspects of an interface metaphor
  - buttons
    - horizontal or vertical
  - boxes

Hierarchical Menus

- most frequently used form of menus



Office Automation

Word Processing

Electronic Mail

Spreadsheet

Graphics

Create

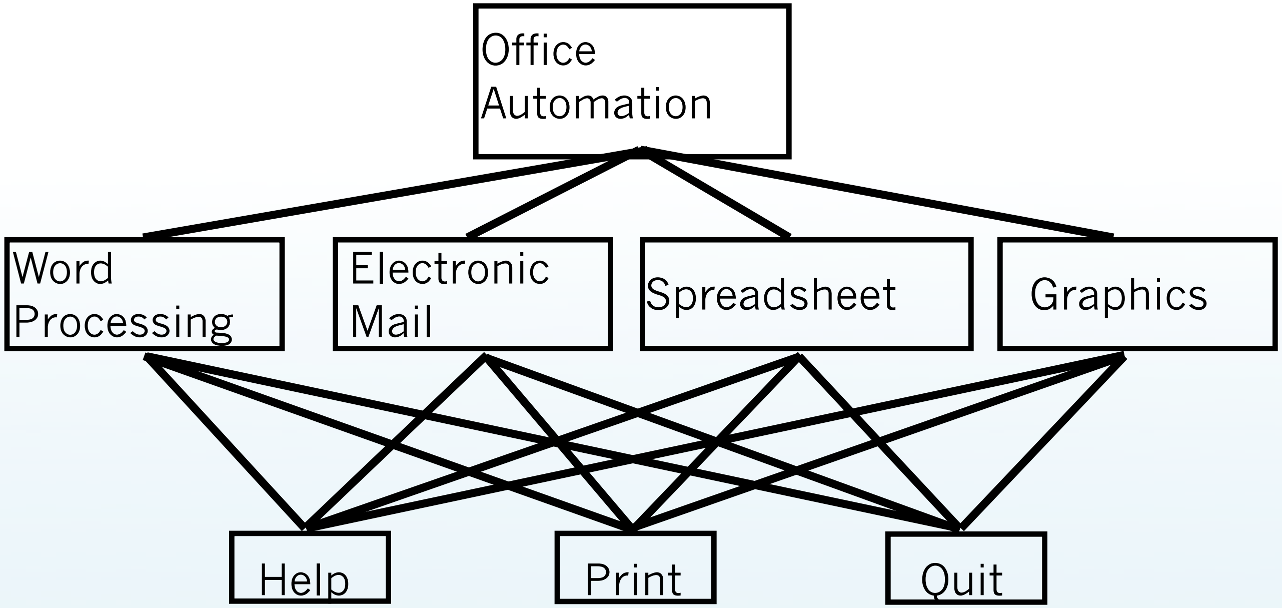
Edit

Print

File

[Mustillo]

Networked Menus



Office Automation

Word Processing

Electronic Mail

Spreadsheet

Graphics

Help

Print

Quit

[Mustillo]

Menu Types

- fixed menus
  - remain in place until the option is selected
- pull-down menus (drop-down menus)
  - dragged down from a menu bar at the top of the screen, an item is selected, and the menu automatically returns back to its original title
- pop-up menus
  - appear when a user clicks on a particular area of the screen, which may be designated by an icon
  - menu remains in place until the user instructs it to disappear, usually by clicking on a "close" box in the border of the menu's window
- cascading menus
  - display all options chosen one after the other in a cascading fashion

[Mustillo]

Design Considerations for Menus

- **order of items**
  - alphabetically, by category, or by frequency
  - some functions should be kept apart (e.g., ‘create’ option should not be placed next to the ‘delete’ option)
- **selection of items**
  - number or letter corresponding to the required option
  - pointing at the option using a pointing device
  - highlighting the item through cursor control keys
- **navigation through a series of menus**
  - hierarchically-structured menus
    - main menu with a series of sub-menus to make further selections

[Mustillo]

© Franz J. Kurfess

Advantages of Menus

- self-explanatory
  - require little training or learning
- require little human memory
  - easy to remember, recognition rather than recall
- indicates when parts are not relevant
- interaction under user control
- fast, efficient interface for experienced users
  - shortcuts to mouse actions
- few keystrokes, which reduces errors
- easy error handling
- enhancements and changes are visible

[Mustillo]

Problems with Menus

- **not appropriate or efficient for some users and tasks**
  - may slow down frequent or expert users
- **users may get lost in menu hierarchies**
- **menu names may not be meaningful to users**
- **inflexible**
  - unless highly networked, the user is forced through set sequences of steps
- **impractical for numerous choices**
  - good for a limited number of valid inputs at any given time
- **use lots of screen space**
- **most effective with a pointing device**
  - mouse, trackball, touch screen

[Mustillo]

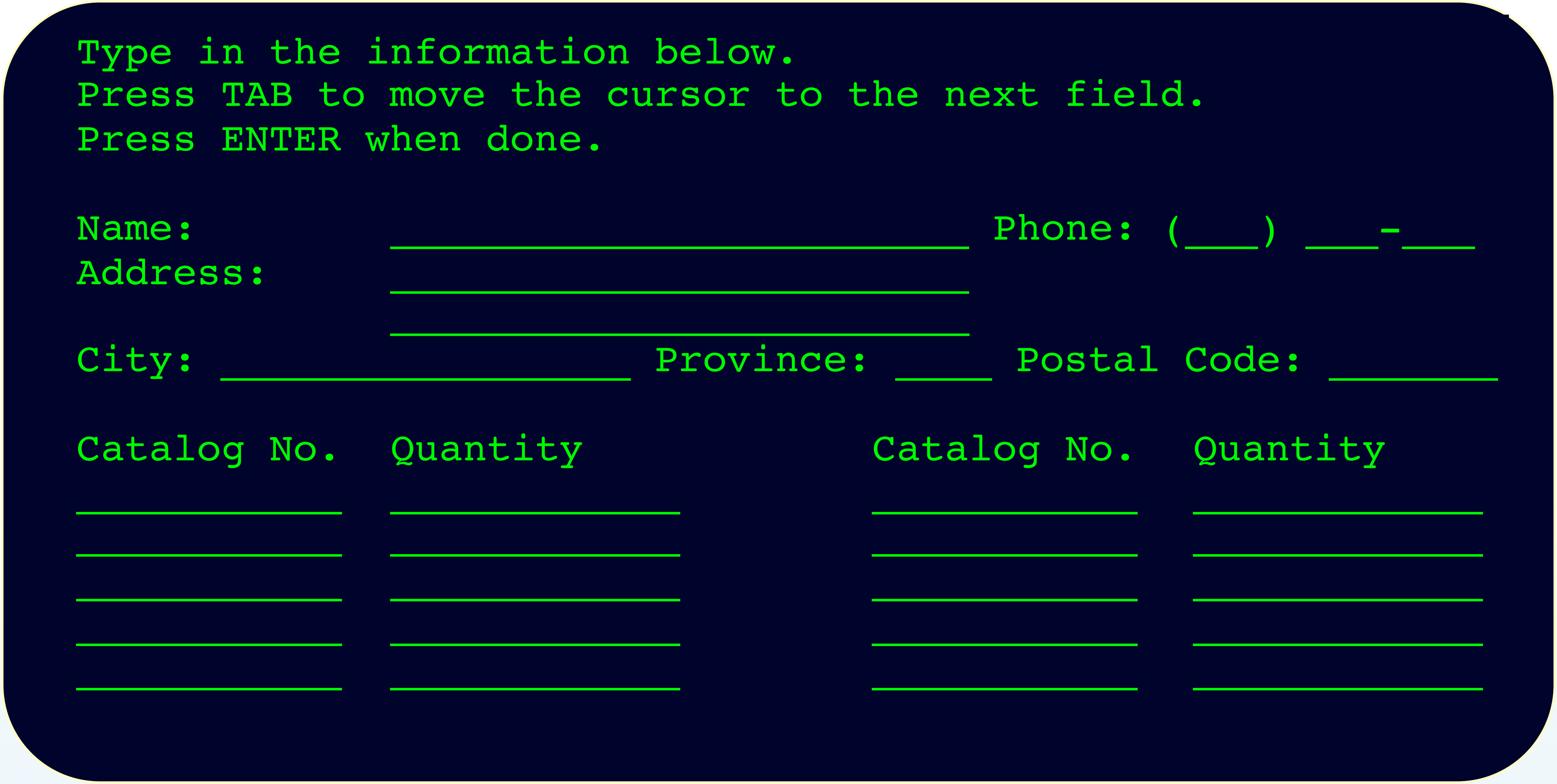
Fill-in Forms

- **similar to a paper fill-in form**
  - presented on a computer screen instead of paper
- **formatted structure containing fields, in which the user inputs data**
  - each field has a label or caption that indicates the type of data to be entered in that field
- **organization and layout are important aspects**

[Mustillo]

© Franz J. Kurfess

Example Fill-in Forms



```
Type in the information below.
Press TAB to move the cursor to the next field.
Press ENTER when done.

Name:         ________________________  Phone: (___) ___-___
Address:      ________________________
              ________________________
City: _________________  Province: ____  Postal Code: _______

Catalog No.  Quantity             Catalog No.  Quantity

___________  ____________         ___________  ____________
___________  ____________         ___________  ____________
___________  ____________         ___________  ____________
___________  ____________         ___________  ____________
___________  ____________         ___________  ____________
```

A fill-in form interface design for a department store

[Mustillo]

Advantages Fill-in Forms

- **self-explanatory**
  - little training or learning required
- **require little human memory**
  - rely on recognition rather than recall
- **efficient use of screen space**
  - multiple fields can be presented on one screen
- **parameters with many possible input values**
  - e.g., names, street names, city names, postal codes, etc.
- **provide context**
  - context information from other fields
- **enhancements and changes are visible**

[Mustillo]

© Franz J. Kurfess

Problems Fill-in Forms

- assume knowledge of valid inputs
  - e.g., catalog numbers, part numbers, etc.
- assume typing skills
- users need to know how to use TAB to move to the next field, BACKSPACE to correct
- resent opportunities for making errors
- inflexible
  - input order cannot be changed

[Mustillo]

Direct Manipulation

- **describes the interaction between user and object**
  - directly connects an action to an observable response from an object
- **follows an object-action paradigm**
  - user performs tasks by selecting an object (e.g., icon, window, or text), and then selects an action (e.g., move, close, underline) to perform on that object
- **permits users to control their environment**
  - by directly manipulating graphical objects and controls similar to those that they encounter in real life
  - push button starts an action
  - slider is used to select an analog setting, etc.
  - mapped onto affordances

[Mustillo]

Principles of Direct Manipulation

- visible objects of interest
  - representing an object with an icon
- continuous representation of objects and actions of interest in a meaningful metaphor
- objects and actions are shown
- rapid, reversible, incremental actions
- immediate visibility of results of actions
- direct manipulation of the object of interest instead of a complex command language syntax
  - typing is replaced with pointing and selecting

[Mustillo]

Indirect Manipulation

- **in many industrial applications, objects on the computer screen stand for devices in the real world**
- **manipulation of the screen objects may not directly constitute an identical manipulation of the real-world objects**
- **feedback is required on two levels**
  - directly from the user interface
  - indirectly, but more relevant from the real-world devices

Direct Manipulation Interfaces

- **powerful**
  - expert users can work quickly, wide range of tasks
- **novice users can learn basic functionality quickly**
  - either through self exploration or through a demonstration by a more experienced user.
  - no need to learn and remember complex commands
- **goal-oriented**
  - users can see immediately if their actions are helping them realize their goals
  - if not, they can simply change the direction of their activity
- **error messages are rarely needed**

[Shneiderman, Mustillo]

WIMP Interfaces

- **direct manipulation interfaces are characterized by**
  - windows
    - to divide the screen into areas
  - icons representing objects
    - can be moved around the screen
  - mouse (or another pointing device)
    - used to manipulate objects on the screen
  - pop-up or pull-down menus
    - display the available options

[Mustillo]

WIMP Advantages

- **system as a whole is easily visible**
  - objects are represented by icons
  - available menu options can be inspected by pulling down menus
- **basic actions are consistent across systems**
  - opening, closing, copying, deleting, scrolling, etc.
  - makes learning easier
- **users are free to explore different aspects of the system**
  - most actions can be ‘undone’ (reversed) if it does not have the desired or required effect

[Mustillo]

© Franz J. Kurfess

GUIs and Direct Manipulation

- **GUIs rely heavily on direct manipulation**
  - have become synonymous with the WIMP interface
- **in addition to having all of the characteristics of DM and WIMP interfaces, GUIs are also strongly graphical**
- **the primary interaction style is direct manipulation**

[Mustillo]

Advantages of Direct Manipulation

- easy to learn and remember
  - rely on recognition, not recall
- comprehensible, predictable, and controllable
  - users experience less anxiety
  - users feel in control
- visual: WYSIWYG
- flexible, reversible actions
  - undo -> you can always choose to de-select a menu item
- provide context, and instant visual feedback
- exploit human use of visual-spatial cues
  - “a picture is worth a thousand words!”
- less prone to errors
  - low typing requirement, visual feedback
  - less need for error messages

[Mustillo]

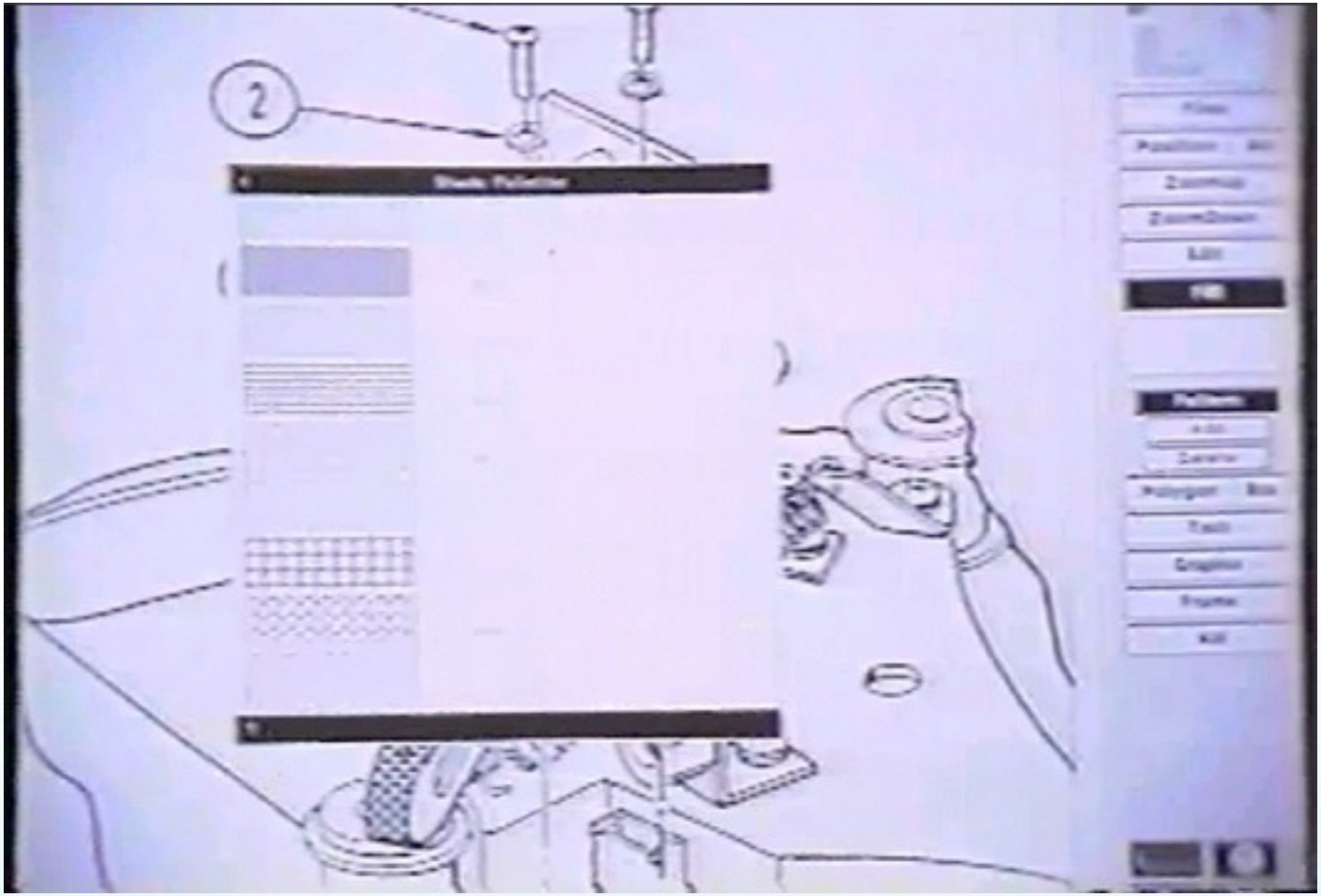
Problems with Direct Manipulation

- not self-explanatory
  - not necessarily intuitive or obvious to first-time users
  - users must learn meaning of the visual representation
- can be inefficient for high-frequency users
  - same tasks much faster/ more efficient with command languages
- moving a mouse and pointing slower than pressing keys
- repetitive tasks
  - users need a macro or scripting mechanism to handle repeated manipulations
- limited accuracy
  - pointer system may not be accurate enough or sufficiently controllable for some tasks

[Mustillo]

GUI Example 1

- **ICL/Three River PERQ workstation**



http://toastytech.com/guis/perqintranvid3.jpg

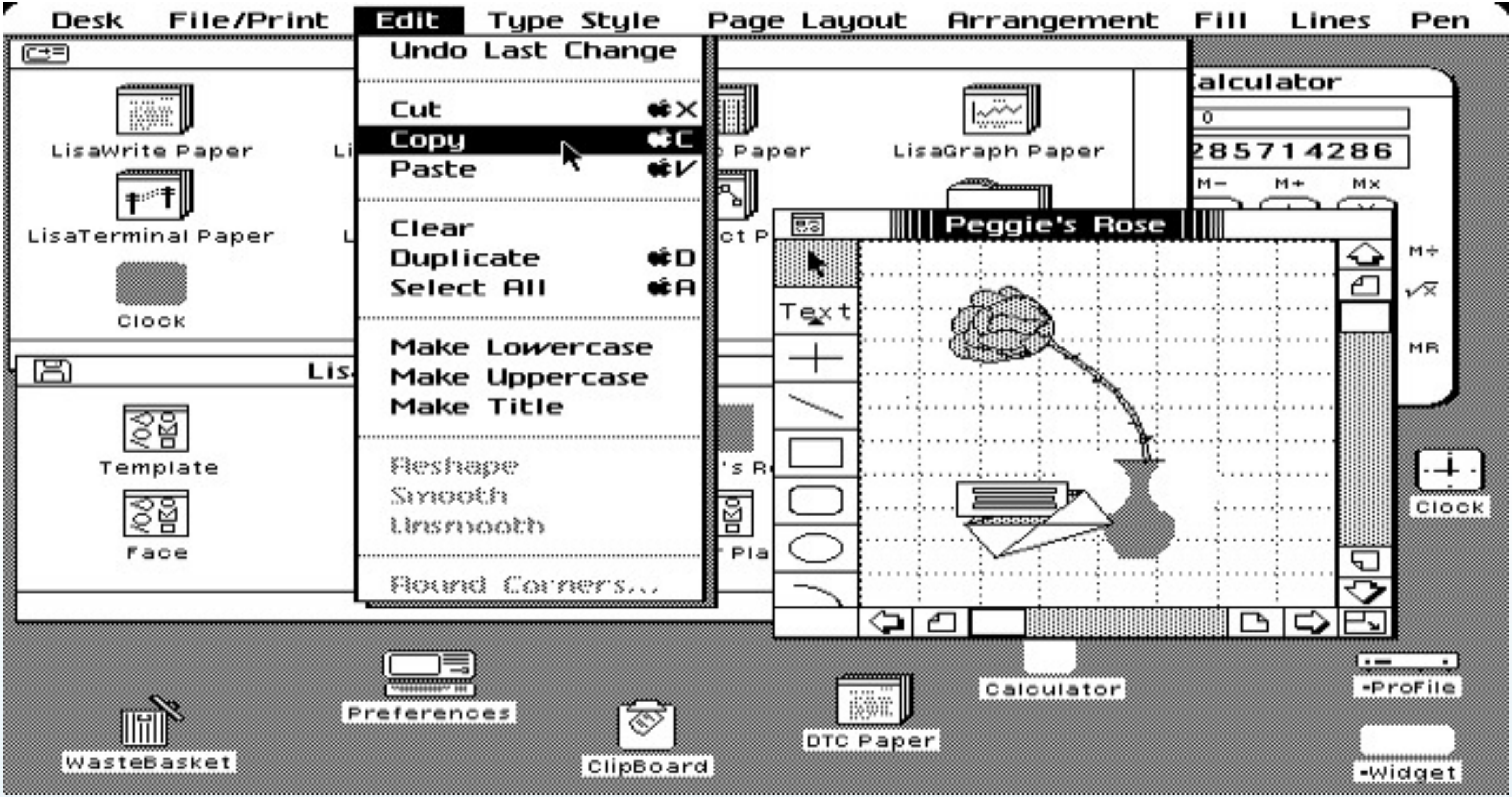
© Franz J. Kurfess

Movie: Advanced GUI, ca 1980

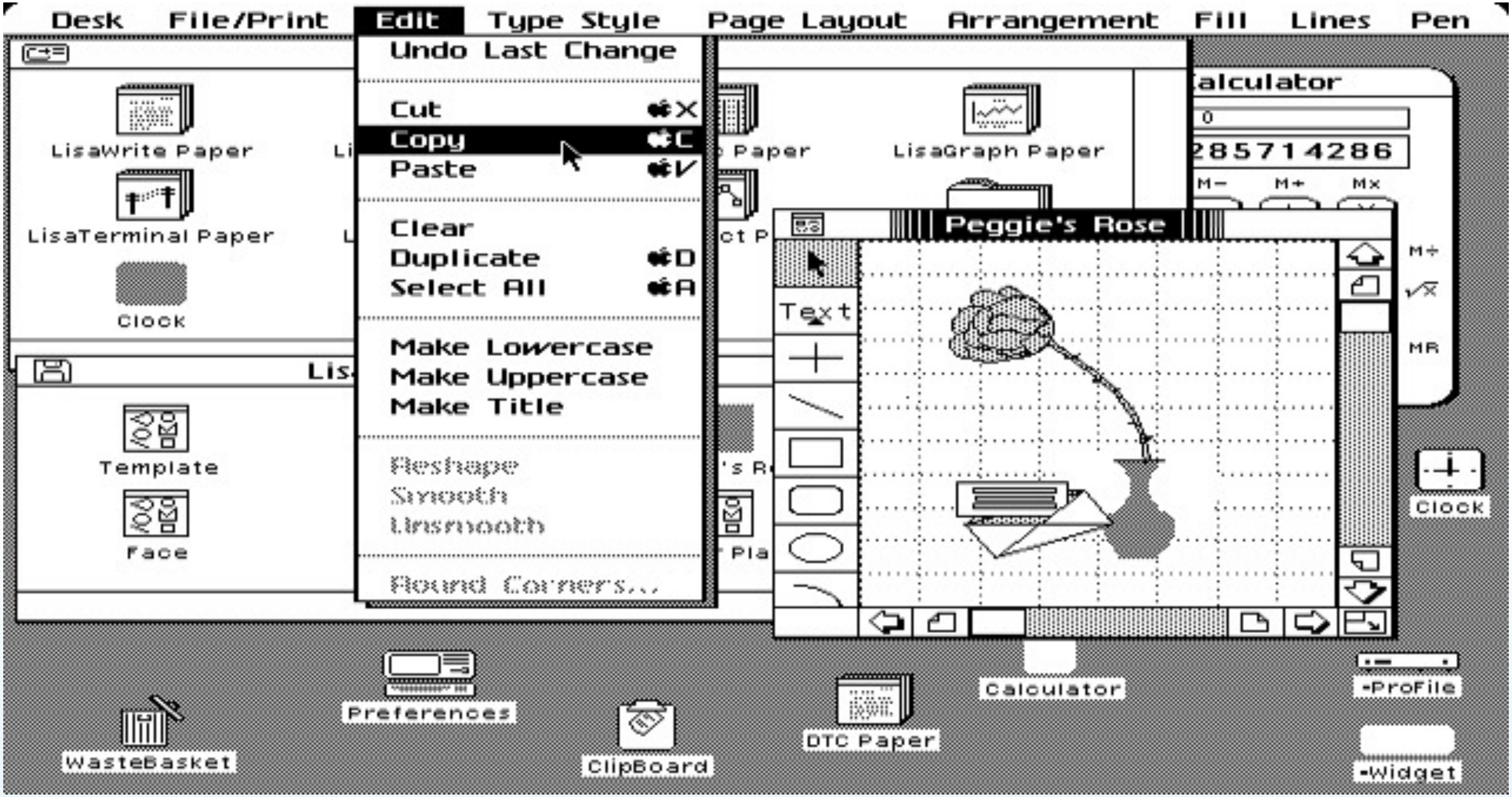
- a marketing movie for the INTRAN software package for scientific design on the PERQ2 workstation
  - http://www.youtube.com/watch?v=Fap-mXY80Is&feature=gv&hl=en
    - computer stuff starts around the 2:00 min mark

© Franz J. Kurfess

GUI Example 2



GUI Example 2

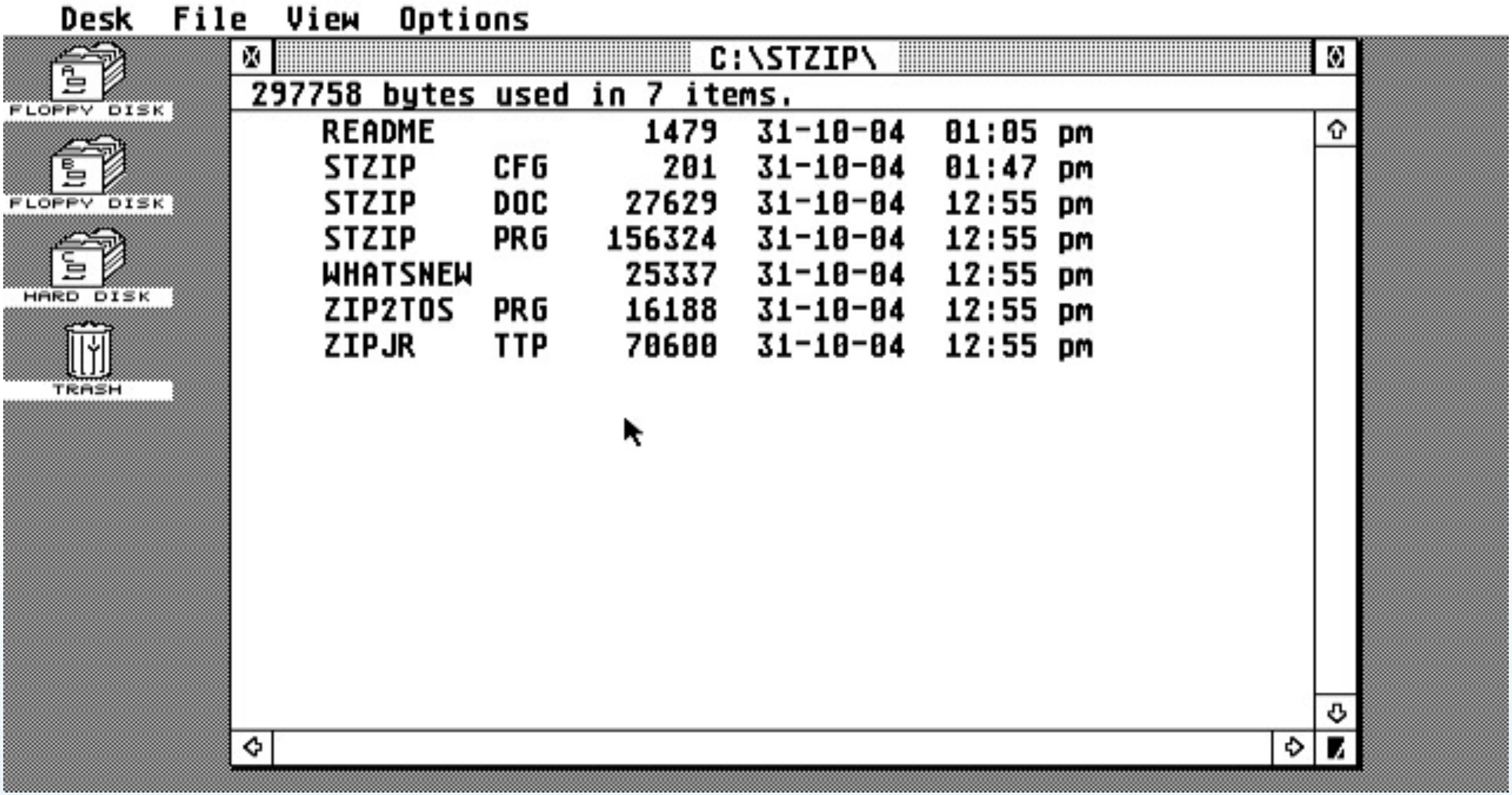


Apple Lisa

http://www.minitrue.nl/essays/vanfresconaarjpeg/graph/lisa2.gif

© Franz J. Kurfess

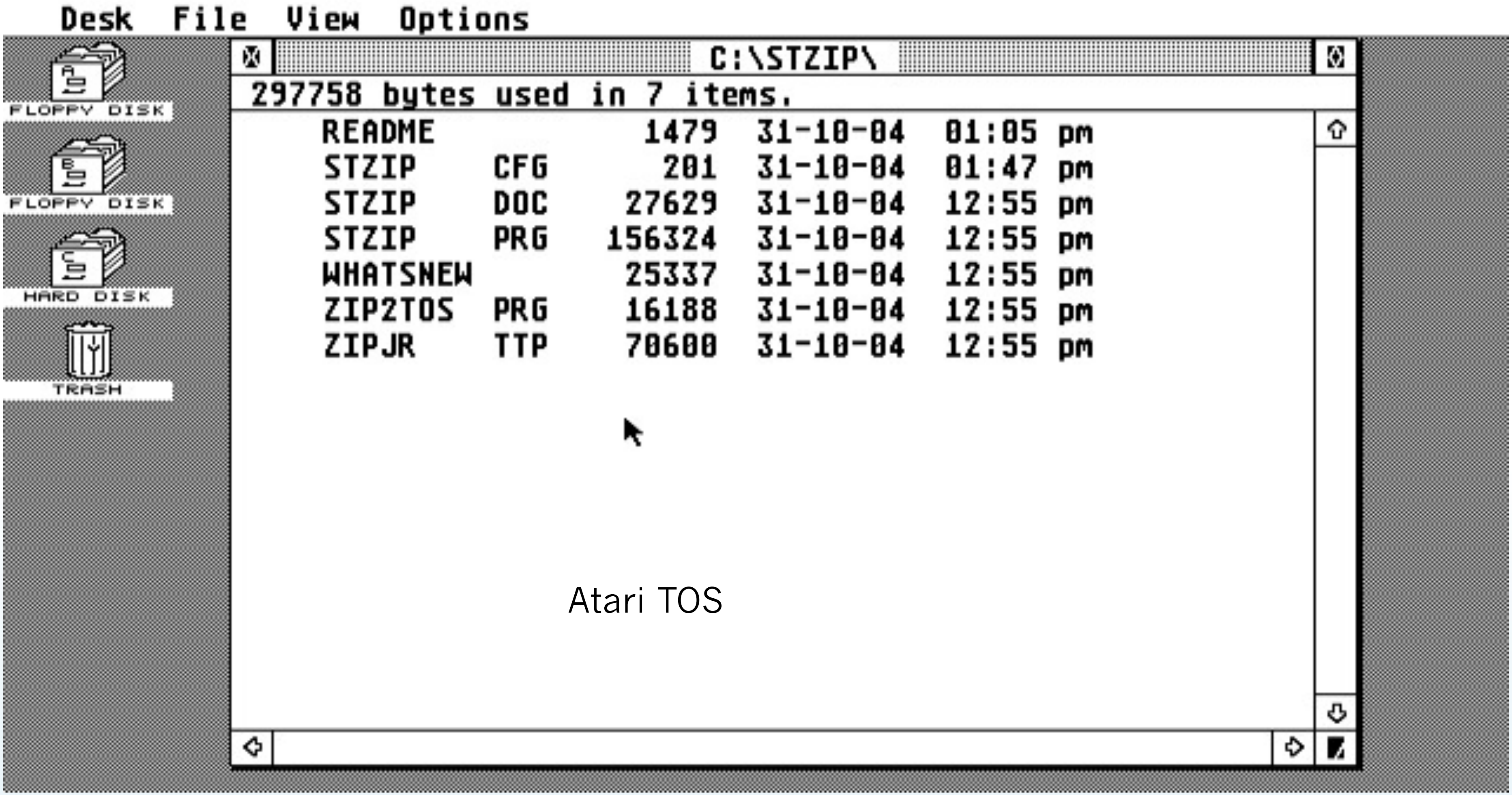
GUI Example 3



```
C:\STZIP\
297758 bytes used in 7 items.
    README               1479  31-10-04  01:05 pm
    STZIP     CFG         201  31-10-04  01:47 pm
    STZIP     DOC       27629  31-10-04  12:55 pm
    STZIP     PRG      156324  31-10-04  12:55 pm
    WHATSNEW            25337  31-10-04  12:55 pm
    ZIP2TOS   PRG       16188  31-10-04  12:55 pm
    ZIPJR     TTP       70600  31-10-04  12:55 pm
```

http://upload.wikimedia.org/wikipedia/en/0/07/ST_Desktop.png

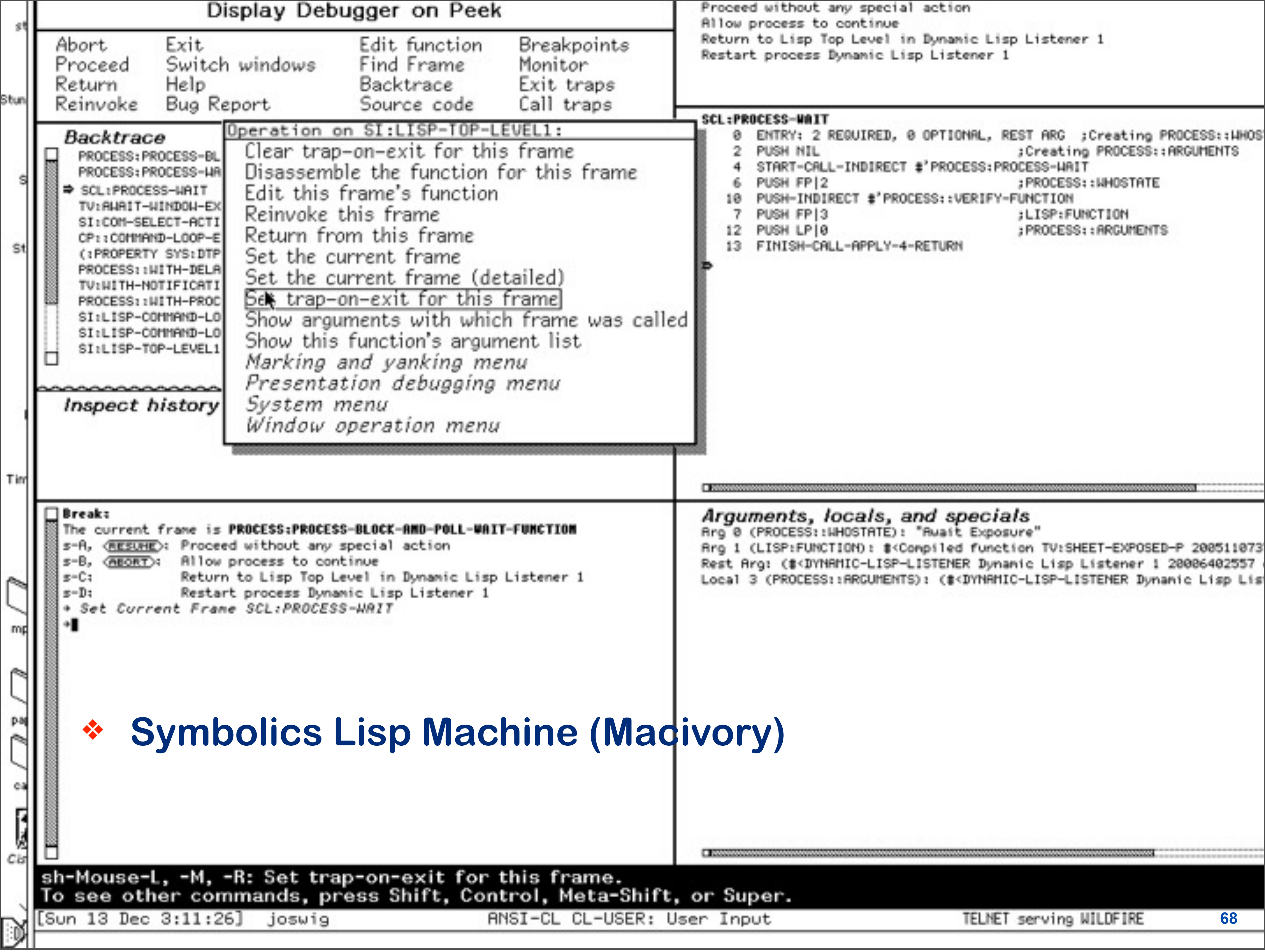
GUI Example 3



```
C:\STZIP\
297758 bytes used in 7 items.
    README              1479  31-10-04  01:05 pm
    STZIP     CFG        201  31-10-04  01:47 pm
    STZIP     DOC      27629  31-10-04  12:55 pm
    STZIP     PRG     156324  31-10-04  12:55 pm
    WHATSNEW           25337  31-10-04  12:55 pm
    ZIP2TOS   PRG      16188  31-10-04  12:55 pm
    ZIPJR     TTP      70600  31-10-04  12:55 pm
```

Atari TOS

http://upload.wikimedia.org/wikipedia/en/0/07/ST_Desktop.png


Display Debugger on Peek
Abort
Proceed
Return
Reinvoke
Exit
Switch windows
Help
Bug Report
Edit function
Find Frame
Backtrace
Source code
Breakpoints
Monitor
Exit traps
Call traps
Backtrace
Inspect history
Operation on SI:LISP-TOP-LEVEL1:
Clear trap-on-exit for this frame
Disassemble the function for this frame
Edit this frame's function
Reinvoke this frame
Return from this frame
Set the current frame
Set the current frame (detailed)
Set trap-on-exit for this frame
Show arguments with which frame was called
Show this function's argument list
Marking and yanking menu
Presentation debugging menu
System menu
Window operation menu
Arguments, locals, and specials

```
Proceed without any special action
Allow process to continue
Return to Lisp Top Level in Dynamic Lisp Listener 1
Restart process Dynamic Lisp Listener 1
```

```
PROCESS:PROCESS-BL
PROCESS:PROCESS-WA
➡ SCL:PROCESS-WAIT
TV:AWAIT-WINDOW-EX
SI:COM-SELECT-ACTI
CP::COMMAND-LOOP-E
(:PROPERTY SYS:DTP
PROCESS::WITH-DELA
TV:WITH-NOTIFICATI
PROCESS::WITH-PROC
SI:LISP-COMMAND-LO
SI:LISP-COMMAND-LO
SI:LISP-TOP-LEVEL1
```

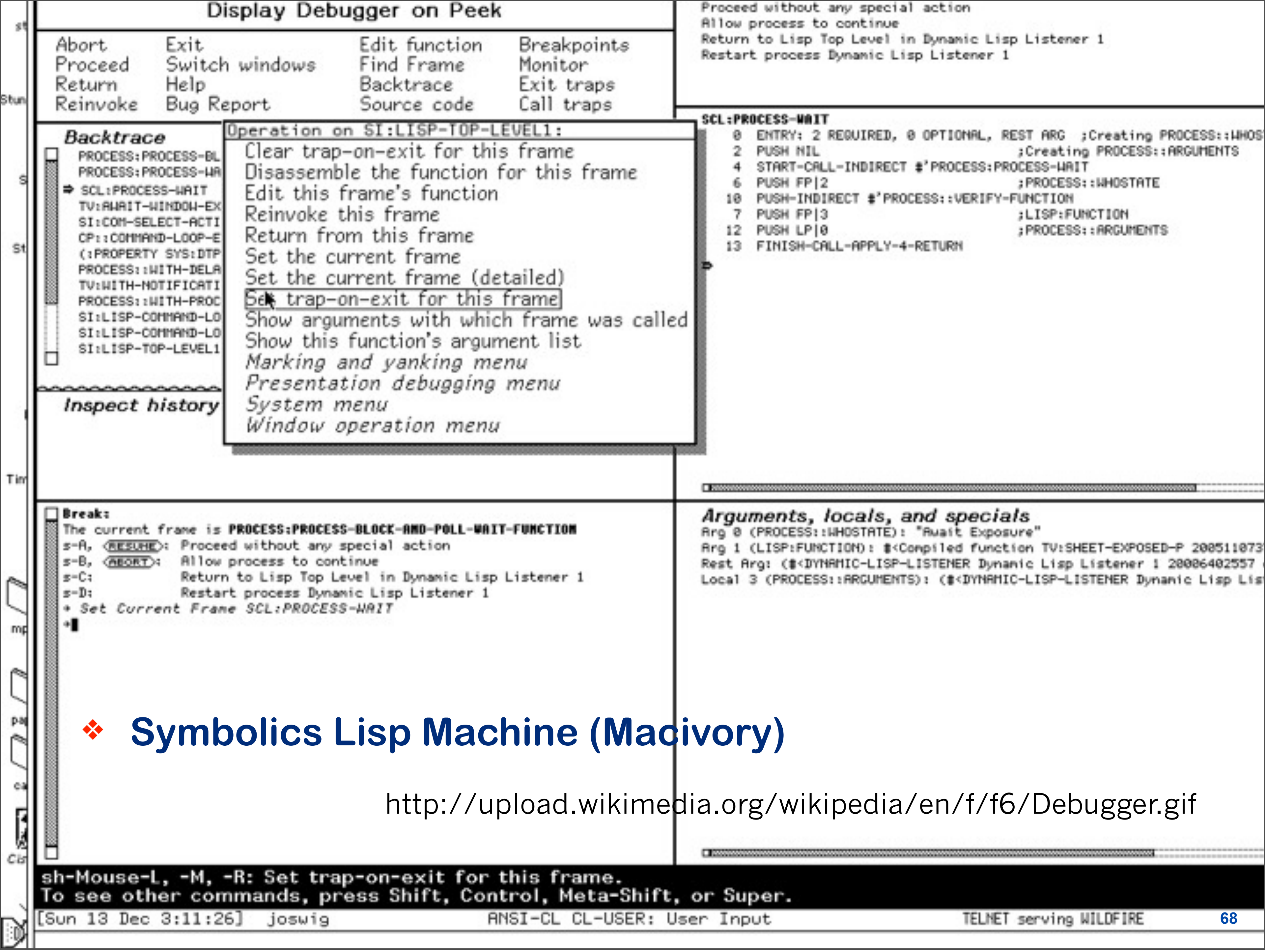
```
SCL:PROCESS-WAIT
  0  ENTRY: 2 REQUIRED, 0 OPTIONAL, REST ARG  ;Creating PROCESS::WHOS
  2  PUSH NIL                                 ;Creating PROCESS::ARGUMENTS
  4  START-CALL-INDIRECT #'PROCESS:PROCESS-WAIT
  6  PUSH FP|2                                ;PROCESS::WHOSTATE
 10  PUSH-INDIRECT #'PROCESS::VERIFY-FUNCTION
  7  PUSH FP|3                                ;LISP:FUNCTION
 12  PUSH LP|0                                ;PROCESS::ARGUMENTS
 13  FINISH-CALL-APPLY-4-RETURN
```

```
Break:
The current frame is PROCESS:PROCESS-BLOCK-AND-POLL-WAIT-FUNCTION
s-A, <RESUME>: Proceed without any special action
s-B, <ABORT>:  Allow process to continue
s-C:           Return to Lisp Top Level in Dynamic Lisp Listener 1
s-D:           Restart process Dynamic Lisp Listener 1
→ Set Current Frame SCL:PROCESS-WAIT
→▮
```

```
Arg 0 (PROCESS::WHOSTATE): "Await Exposure"
Arg 1 (LISP:FUNCTION): #<Compiled function TV:SHEET-EXPOSED-P 20051107
Rest Arg: (#<DYNAMIC-LISP-LISTENER Dynamic Lisp Listener 1 20006402557
Local 3 (PROCESS::ARGUMENTS): (#<DYNAMIC-LISP-LISTENER Dynamic Lisp Lis
```

```
sh-Mouse-L, -M, -R: Set trap-on-exit for this frame.
To see other commands, press Shift, Control, Meta-Shift, or Super.
[Sun 13 Dec 3:11:26]   joswig          ANSI-CL CL-USER: User Input          TELNET serving WILDFIRE
```

- Symbolics Lisp Machine (Macivory)


Display Debugger on Peek
Abort
Proceed
Return
Reinvoke
Exit
Switch windows
Help
Bug Report
Edit function
Find Frame
Backtrace
Source code
Breakpoints
Monitor
Exit traps
Call traps
Backtrace
Operation on SI:LISP-TOP-LEVEL1:
Clear trap-on-exit for this frame
Disassemble the function for this frame
Edit this frame's function
Reinvoke this frame
Return from this frame
Set the current frame
Set the current frame (detailed)
Set trap-on-exit for this frame
Show arguments with which frame was called
Show this function's argument list
Marking and yanking menu
Presentation debugging menu
System menu
Window operation menu
Inspect history
Arguments, locals, and specials

```
Proceed without any special action
Allow process to continue
Return to Lisp Top Level in Dynamic Lisp Listener 1
Restart process Dynamic Lisp Listener 1
```

```
PROCESS:PROCESS-BL
PROCESS:PROCESS-WA
⇒ SCL:PROCESS-WAIT
TV:AWAIT-WINDOW-EX
SI:COM-SELECT-ACTI
CP::COMMAND-LOOP-E
(:PROPERTY SYS:DTP
PROCESS::WITH-DELA
TV:WITH-NOTIFICATI
PROCESS::WITH-PROC
SI:LISP-COMMAND-LO
SI:LISP-COMMAND-LO
SI:LISP-TOP-LEVEL1
```

```
SCL:PROCESS-WAIT
  0  ENTRY: 2 REQUIRED, 0 OPTIONAL, REST ARG  ;Creating PROCESS::WHOS
  2  PUSH NIL                                 ;Creating PROCESS::ARGUMENTS
  4  START-CALL-INDIRECT #'PROCESS:PROCESS-WAIT
  6  PUSH FP|2                                ;PROCESS::WHOSTATE
 10  PUSH-INDIRECT #'PROCESS::VERIFY-FUNCTION
  7  PUSH FP|3                                ;LISP:FUNCTION
 12  PUSH LP|0                                ;PROCESS::ARGUMENTS
 13  FINISH-CALL-APPLY-4-RETURN
```

```
Break:
The current frame is PROCESS:PROCESS-BLOCK-AND-POLL-WAIT-FUNCTION
s-A, RESUME:  Proceed without any special action
s-B, ABORT:   Allow process to continue
s-C:          Return to Lisp Top Level in Dynamic Lisp Listener 1
s-D:          Restart process Dynamic Lisp Listener 1
→ Set Current Frame SCL:PROCESS-WAIT
→▮
```

```
Arg 0 (PROCESS::WHOSTATE): "Await Exposure"
Arg 1 (LISP:FUNCTION): #<Compiled function TV:SHEET-EXPOSED-P 20051107
Rest Arg: (#<DYNAMIC-LISP-LISTENER Dynamic Lisp Listener 1 20006402557
Local 3 (PROCESS::ARGUMENTS): (#<DYNAMIC-LISP-LISTENER Dynamic Lisp Lis
```

```
sh-Mouse-L, -M, -R: Set trap-on-exit for this frame.
To see other commands, press Shift, Control, Meta-Shift, or Super.
[Sun 13 Dec 3:11:26]   joswig          ANSI-CL CL-USER: User Input          TELNET serving WILDFIRE
```

- ❖ **Symbolics Lisp Machine (Macivory)**

http://upload.wikimedia.org/wikipedia/en/f/f6/Debugger.gif

Intelligent Agents

- user conveys intentions
  - goals
  - critical parameters
- computer handles mundane and tedious activities
  - repetitive
  - easy to automate
- less, but more effective interaction
- computer acts more autonomously

Mobile Devices

- Usage
- Capabilities
- Advantages
- Limitations

Mobile Devices - Usage

- **often closer proximity to users**
  - with the user most of the time
  - easy to carry
- **often multi-purpose devices**
  - mobile phone, music player, camera, hand-held computer
- **essential professional or personal device**
  - connectivity (phone, text messaging, email, Web)
  - organization (calendar, to do list, contacts, directions)
  - pleasure (music, photos, videos, e-books)

Mobile Devices - Capabilities

- **I/O capabilities**
  - input
    - control and navigation (buttons; no mouse; cursor keys or limited pointing device; touch screen; gyroscope or accelerometer)
    - text (keyboard missing or small)
    - speech (microphone/head set)
  - output
    - visual: small screen
    - audio: small speaker, headphones
    - haptic: vibration
- **computational capabilities**
  - limited memory, processing
- **connectivity**
  - wired (USB)
  - wireless (cellular, Wi-Fi, Bluetooth, Infrared)

© Franz J. Kurfess

Mobile Devices - Advantages

- quick & easy access
- multiple functions in one device

© Franz J. Kurfess

Mobile Devices - Limitations

- **input and output constraints**
  - buttons, keyboard, navigation
  - screen size
- **functional constraints**
  - available functions are often not very sophisticated
- **proprietary or unusual interaction methods**
  - touch gestures, (virtual) keyboard arrangement
- **synchronization with other devices**
  - computer, home phone, car, ...

Ubiquitous and Pervasive Computing

- Usage
- Capabilities
- Advantages
- Limitations

Ubiquitous Computing – Usage

- **task-specific devices with computing capabilities**
  - entertainment systems
  - home security
  - phone
  - car navigation systems
- **core functionality augmented by computational capabilities**
  - connectivity
  - additional functionality
  - better interaction

Ubiquitous Computing - Capabilities

- **core capabilities defined by the application are or task**
  - often very limited
  - sometimes augmented for better functionality or interaction
- **Input**
  - often buttons, remote control
- **Output**
  - screen, audio, actuators (performing specific functions)
- **computational capabilities**
  - often embedded microprocessors

Ubiquitous Computing - Advantages

- familiar devices and tasks
- added functionality for such devices
- better interaction with devices
  - e.g. Tivo vs. VCR
- connectivity between computers and household or personal devices

© Franz J. Kurfess

Ubiquitous Computing - Limitations

- limited computational capabilities
- limited interaction methods
- coordination and synchronization

Important Concepts and Terms

- batch system
- command-line interface
- contextual task analysis
- desktop
- direct manipulation
- forms
- full-screen interface
- goal
- graphical user interface (GUI)
- heuristic evaluation
- hierarchical menu
- human-machine interface
- intelligent agent
- interaction style
- menu
- mouse
- natural language
- networked menu
- system language
- task
- task analysis
- usability
- user-centered design
- user interface design
- user language
- user requirements
- What You See Is What You Get" (WYSIWYG)
- WIMP
- window

Chapter Summary

- **practically all current interaction styles are variations of command-based interfaces**
- **the currently predominant interaction style is the WIMP interface**
  - windows, icons, mouse, pull-down menus
- **the selection among various possible interaction styles depends on user background, task requirements, technology, and economical factors**
- **emerging technologies like natural language processing or intelligent agents may cause a shift towards non-command interfaces**

© Franz J. Kurfess