# A Java Implementation of a Kaypro II Microcomputer System

A CSI-426 Senior Project

Dr. Charles Howerton, Instructor

Presented By:

Joe Diethorn

**Shannon Steinmetz** 

nternet address: www.yoy.org/kaypro, Email: kaypro@yoy.org

### **Team Members**

- Team members
  - Brandon Buist
  - **♦** Joe Diethorn
  - **♦ Jim Gilmer**
  - **♦** Shannon Steinmetz
  - **♦ Hung To**
- Examined strengths of each member
- Tasks assigned according strengths
- All team members participated in design

# **Project Description**

#### The Assignment

- Kaypro II emulator
  - **♦** Debugger
  - **♦** Printer emulation
  - Floppy drive emulation
  - **♦** Utilize CP/M operating system
  - **♦** Runs actual programs under emulated OS
- Implemented as a Java applet (needed to learn)
- Must implement good engineering processes
- Object oriented design and implementation



- Internet
  - **♦** News groups
  - **◆ Contacts**
  - **♦** Schematics
  - **◆ Data books**
- Obtained actual system
  - **♦ Kaypro manuals**
  - **♦ ROM images**
  - **♦ Floppy images**
  - **♦** Benefit of using actual system

# Requirements Gathering

#### **Hardware Emulation Description**

- What is hardware emulation?
  - **♦ Simulate hardware components** 
    - \* CPU
    - ⋆ Floppy controller chip
    - ⋆ Memory mapping
    - ⋆ I/O chips
    - ⋆ Video circuitry
  - Benefits
    - **★ Would run actual software**
    - **★** Greater compatibility

# Requirements Gathering

#### **Hardware Emulation Challenges**

- Needed to extract system ROM's
  - ◆ EPROM reader
  - Converted into static program tables
  - Emulation booted with actual boot ROM code
- Needed to extract floppy images
  - Obtained software that extracted raw disk images
  - Converted into static tables
  - Emulation reads and writes actual Kaypro disk data



Hardware Emulation Challenges Cont...

- Needed to decipher schematics
  - Used data books to define hardware interfaces
  - Used schematics to define system architecture
- Debugging emulation code
  - Needed to create system debugger

### **Documentation**

- User manual
  - **♦** User instructions
  - **♦** Written before implementation
- Requirements definition
  - **♦** Contract with customer (instructor)
  - **♦** High level definition
- Requirements specification
  - **◆** Contract with programmer
  - **◆** Detailed definition
- Design
  - Previous documents describe what is needed
  - **◆** Design describes how to implement



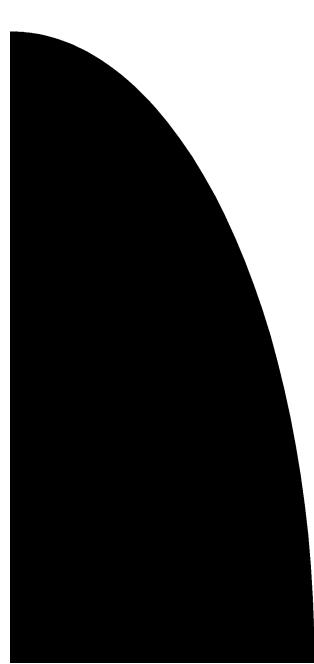
- Test plan
  - **♦** What will be tested and how
- Accumulated information and data
  - **♦ ROM images**
  - **♦ CP/M disassembly**
  - **♦** Kaypro technical manual
- Beta test results
  - **♦** System disclosed to internet community
  - **♦** Since it was an Java applet, internet users could use
  - **◆** Comments were gathered



- Problems and inconstancies were "squeezed" out
- Team was united
- Many "heads" concentrated on complex issues
- Problems were found ahead of implementation
- Coordination between team members
- Incremental integration
- Shortened implementation cycle



- Advantages
  - **◆ Up-front documentation made** implementation easy
  - **◆ Integration was straight forward**
  - **♦** System "fell together"
  - **◆** System worked within one hour of integration
- Unit testing
  - **◆** Each member tested their own code before it was implemented
  - **◆ Documentation made it easy to unit test code**
  - **◆** Good unit testing resulted in successful integration and bug-free operation



# **Testing**

- Executed test plan
- Ran actual programs
- Found actual Kaypro II bugs
  - **◆ Floppy drive selection bug**

## Conclusion

- Benefits of properly engineered software
  - ◆ Properly engineered product yields seamless integration and compatibility and functionality
- Real world project
  - ◆ Far too many schools teach software engineering without exposing the student to a "real" project
  - **♦ The** focus is often on implementation not on the process

## **Conclusion Cont...**

- The power is in the process
  - ◆ Schools need to understand it. Companies need to understand it. The power is in the process. A well-engineered product will last longer, exhibit fewer bugs, come together more smoothly, and result in an empowered, energized design and implementation team
- Our team
  - **♦ The result is total buy-in and total team kmowledge**
  - **♦ Team** was excited and empowered
  - ◆ The software reflects the team that designed and implemented it