Information Systems
ISM 3011

Unit 4A

Principles and Learning Objectives

• When selecting an operating system, you must consider the current and future needs for application software to meet the needs of the organization. In addition, your choice of a particular operating system must be consistent with your choice of hardware.
  – Identify & describe the functions of the two basic kinds of software.
  – Outline the role of the operating system & identify the features of several popular operating systems.

Principles and Learning Objectives

• Do not develop proprietary application software unless doing so will meet a compelling business need that can provide a competitive advantage.
  – Discuss how applications software can support personal, workgroup, and enterprise business objectives.
  – Identify 3 basic approaches to developing applications software and discuss the pros and cons of each.

Principles and Learning Objectives

• Choose a programming language whose functional characteristics are appropriate to the task at hand, taking into consideration the skills and experience of the programming staff.
  – Outline the evolution of programming languages.
• The software industry continues to undergo constant change; users need to be aware of recent trends and issues to be effective in their business and personal life.
  – Differentiate among the five generations of programming languages.

Flextronics /QuoteWin

• What is the problem with QuoteWin from a supplier’s perspective?

Software: Systems and Application Software

Chapter 4
An Overview of Software

- **Computer program** - sequences of instructions for the computer
- **Documentation** - describes program functions
- **Systems software** - coordinates the activities of hardware & programs
- **Applications software** - helps users solve particular problems

The Importance of Software in Business

Operating Systems

- Perform common hardware functions
- Provide a user interface
- Provide hardware independence
- Manage system memory
- Manage processing
- Control access to system resources
- Manage files

Operating Systems

- KERNEL

The Role of the Operating System

Hardware Independence

Examples:
- Accessing Printers (Windows-API) or
- Scanners and Digital Cameras (TWAIN)
Memory Management

Find and reserve free blocks etc.

1 2 3 4 5 6
Memory segments

Processing Tasks

- **Multitasking** - more than one program (task) can run at a time using a single processor
- **Multi-User OS** - multiple users can simultaneously use the resources of a single processor
- **Scalability** - easy adaptation to more users or tasks

System Speed and RAM

- Why does a bigger primary storage capacity (more RAM) increase your PC’s speed?
- Does more memory always lead to a higher system performance?

Off-the-Shelf Software

- The book says on page 147 that one should check whether the software manufacturer is financially solvent and reliable. Why?

Enterprise Operating Systems

- IBM’s OS/390
- HP’s MPE/iX (Multiprogramming Executive with integrated POSIX)
- IBM’s z/OS
- Linux

Sources of Software
Proprietary vs. Off-the-Shelf Software

<table>
<thead>
<tr>
<th></th>
<th>Proprietary Software</th>
<th>Off-the-Shelf Software</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advantages</td>
<td>You can generate what you need, when you need it.</td>
<td>It can take a long time and requires purchased software.</td>
</tr>
<tr>
<td></td>
<td>It can be expensive.</td>
<td>This initial cost is lower, but ongoing maintenance and support costs can be high.</td>
</tr>
<tr>
<td>disadvantages</td>
<td>Being involved in the development costs is a further cost of control over the software.</td>
<td>It requires the purchase of software that may not be compatible with existing systems.</td>
</tr>
<tr>
<td></td>
<td>There is more flexibility in modifying modules that may not be required.</td>
<td>Functionality may be limited to the software provided.</td>
</tr>
</tbody>
</table>
|                      | More functionality is available. | There is a lower risk that the software will fail to meet the basic business needs.

What is Integrated Software?

- **Data Integration**: "One fact at one place" (CODD)
- **Functionality Integration**: one function can interact with others (copy and paste inside one program)
- **Application Integration**: Two or more software applications can interact.
- **Process Integration**: Two or more business processes are connected.

Compiler

Object-Oriented Programming Languages

- **Objects**: data and actions that can be performed on the data
- **Encapsulation**: group items into an object
- **Polymorphism**: one procedure can work with multiple objects
- **Inheritance**: an object in a particular class gets attributes of that class

Thank you!

Any questions? Please send an email to mhepp@computer.org!

http://ruby.fgcu.edu/courses/mhepp/