

PART XV

Other Topics

**(Web technologies: HTTP, CGI,
Java applets; Middleware)**

World Wide Web

- Major application protocol used on the Internet
- Simple interface
- Two concepts
 - Point
 - Click

Web Components

- Browser
- Web Server
- Hypermedia links
- Document representation
- Transfer protocol

Browser

- Application program
- User's interface to Web
- Becomes Web client to fetch information from Web server
- Displays information for user

Web Server

- Running program
- Stores set of Web documents
- Responds to request from browser by sending copy of document

Hypermedia Concept

- Web document contains mixture of
 - Text
 - Images
 - Selectable pointers to other Web pages
- Known as *hypermedia*

Hypermedia Link On A Web Document

- Associated with object or area on screen
- Internally like a symbolic link
- Advantage
 - Can reference document on another computer
- Disadvantage
 - Can become invalid

Web Document

- Called a *Web page*
- One Web page per file
- Can contain
 - Binary image
 - Text file
- Text standard
 - Readable representation ASCII
 - Specifies contents and layout
 - Known as *HyperText Markup Language (HTML)*

Terminology

- Markup language
 - Gives general layout guidelines
 - Does not specify exact placement or format

Consequence Of Using A Markup Language

Web documents use the HyperText Markup Language representation. Instead of specifying a detailed document format, HTML allows a document to contain general guidelines for display, and allows a browser to choose details. Consequently, two browsers may display an HTML document differently.

HTML Details

- Document is free-format
- Embedded *tags* give display guidelines
- Tags often appear in pairs
- Tag format
 - Beginning tag

```
<TAGNAME>
```

- Ending tag

```
</TAGNAME>
```

General Form Of HTML Document

```
<HTML>
```

```
  <HEAD>
```

```
    <TITLE>
```

```
      text that forms the document title
```

```
    </TITLE>
```

```
  </HEAD>
```

```
  <BODY>
```

```
    body of the document appears here
```

```
  </BODY>
```

```
</HTML>
```

Document Format

- HTML source is free-form
- Previous example equivalent to this

```
<HTML><HEAD><TITLE>text that forms  
the document  
title</TITLE></HEAD><BODY>body of the  
document appears here</BODY></HTML>
```

Example HTML Tags

- Begin paragraph

`<P>`

- Line break (force a new line)

`
`

- Main heading (largest, boldest font)

`<H1> ...text... </H1>`

- Next heading (next largest)

`<H2> ...text... </H2>`

Example Of HTML Line Break

- Input

```
Hello there.<BR>This is  
an example<BR>of HTML.
```

- Output

```
Hello there.  
This is an example  
of HTML.
```

Example Of HTML Line Break (continued)

- Input

```
Hello there.<BR><BR>This  
shows<BR> HTML spacing.
```

- Output

Hello there.

This shows
HTML spacing.

Example Of HTML Headings

- Input

```
Hello.<BR><H1>This Is A Heading</H1><BR>Back to  
normal.
```

- Output

Hello.
This Is A Heading
Back to normal.

Other HTML Features

- Numbered or unnumbered lists
- Images
- Links to other pages

Images In HTML

- Explicitly denoted as image
- Specified with *image tag*
- Can specify alignment with text
- Example image tags

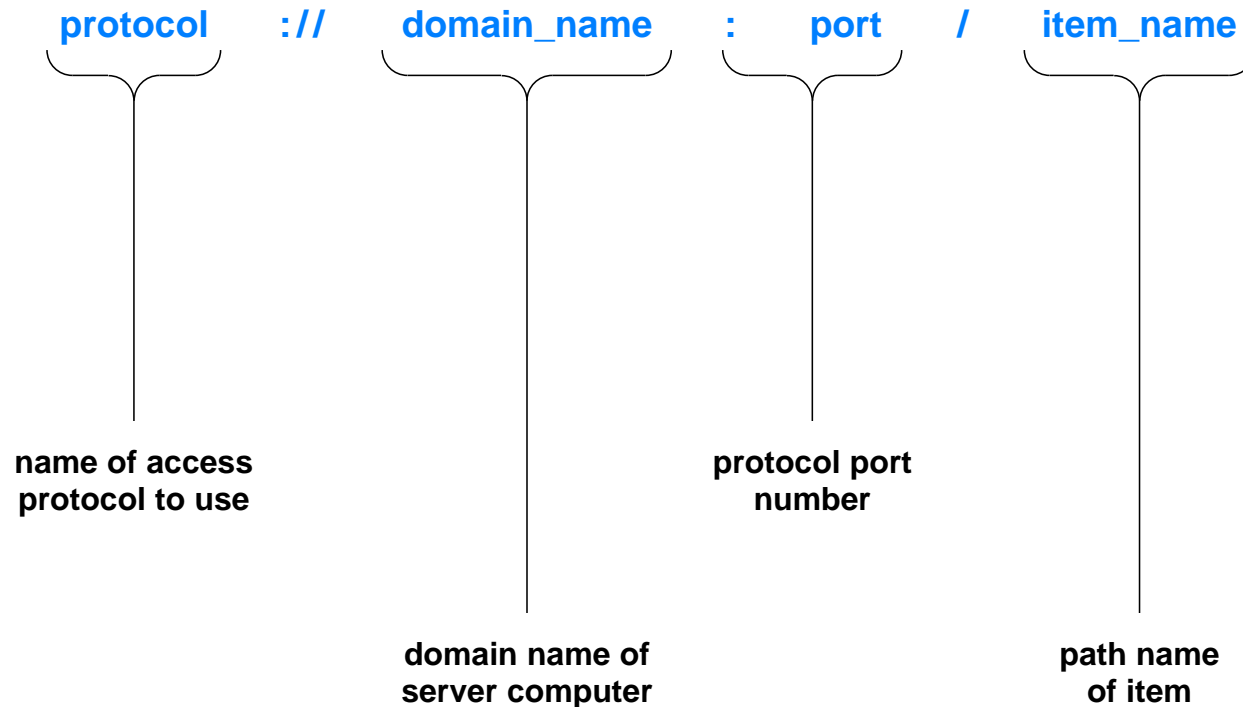
```
<IMG SRC="file_name">
```

```
<IMG SRC="file_name" align=middle>
```

Links To Other Pages

- Symbolic representation
- Embedded in HTML document
- Browser
 - Hides text of link from user
 - Associates link with item on page
 - Makes item selectable
- Called *Uniform Resource Locator (URL)*

General Form Of URL



- Only domain name required
- Defaults
 - Protocol is *http*
 - Port is *80*
 - Path is *index.html*

Link In HTML

- Link specified in `<A>` tag
- Applies to successive items
- Ends with ``
- Called *anchor*

Example Of Anchor Tag In HTML

- Input

The text is published by

Prentice Hall, one of the
larger publishers of Computer
Science textbooks.

- Produces

The text is published by Prentice Hall, one of the larger publishers
of Computer Science textbooks.

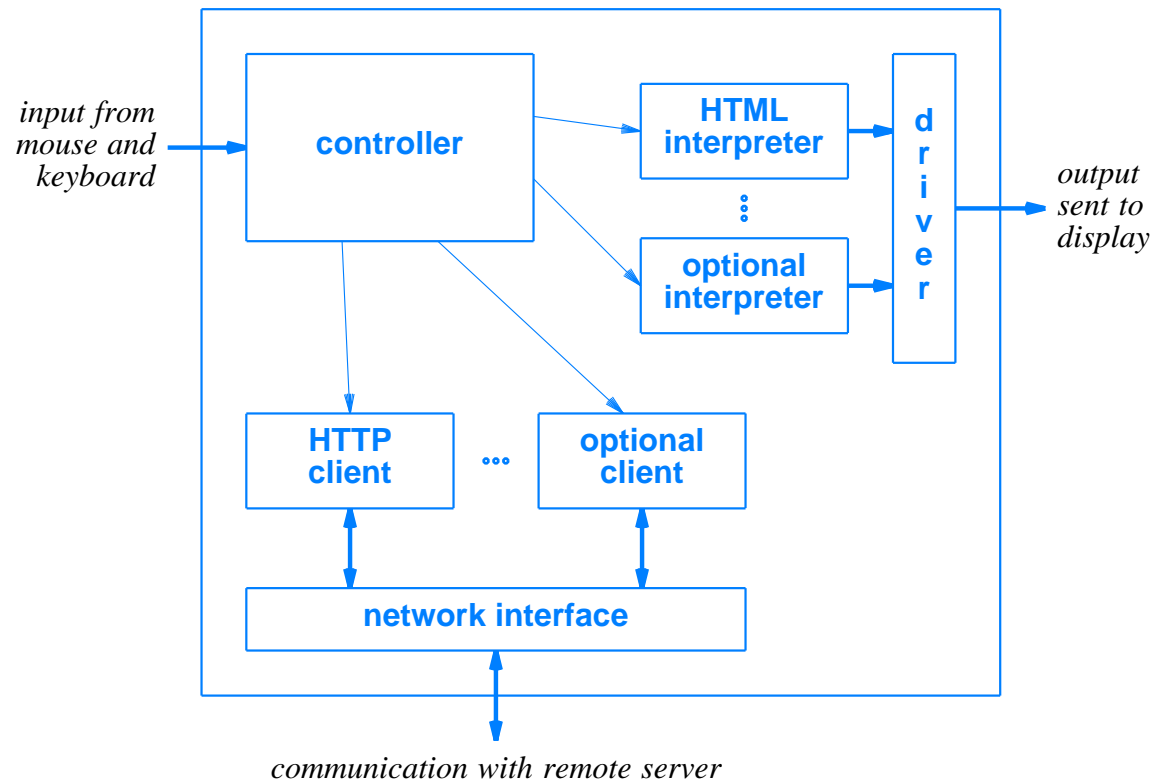
Use Of Client-Server Paradigm

- Web server
 - Makes set of pages available
 - Uses port 80
- Web client
 - Called a *browser*
 - Creates TCP connection to server
 - Sends requests for items
- Primary protocol known as *HyperText Transfer Protocol (HTTP)*

Inside A Browser

- Main controller
 - Receives input from user
 - Invokes client and interpreter
- Client
 - One or more built into browser
 - Uses network to fetch items
- Interpreter
 - One or more built in
 - Displays items

Illustration Of A Browser



- Browser contains many components

Alternative Protocol Example

- File transfer service
- Protocol is FTP
- Example URL

`ftp://ftp.cs.purdue.edu/pub/comer/netbook/client.c`

- Can be used in anchor tag

Caching In Browsers

- Cache for recently accessed
 - HTML pages
 - Images
- Item normally fetched from cache
- User can override
- HTTP can verify timestamp before fetching new copy

Types Of Web Pages

- Static
 - Stored in file
 - Unchanging
- Dynamic
 - Formed by server
 - Created on demand
 - Output from a program
 - Use *Common Gateway Interface (CGI)* technology

Types Of Web Pages (continued)

- Active
 - Executed at client
 - Consist of a computer program
 - Can interact with user
 - Use *Java* technology

Summary Of Web Document Types

Web documents can be grouped into three categories depending on when the information in the document changes. The information in a static document remains unchanged until the author revises the document. The information in a dynamic document can change whenever a server receives a request for the document. Information displayed by an active document can change after the document has been loaded into a browser.

CGI Technology

- URL specifies
 - Location of Web server
 - CGI program on that server
 - Arguments to program
- Web server
 - Uses TCP for communication
 - Accepts HTTP request from client
 - Runs specified CGI program
 - Returns output to client

CGI Technology

(continued)

- CGI program
 - Performs arbitrary computation
 - Often written in a scripting language
 - Produces output file when run
 - Starts output with header

Header In CGI Output

- Stops at first blank line
- Identifies
 - Encoding used
 - Type of document
- Format

Keyword : information

CGI Header Examples

- HTML document header

`Content Type: text/html`

- Text document header

`Content Type: text/plain`

- Redirection header

`Location: /over_here/item4`

Example CGI script

```
#!/bin/sh

#
# CGI script that prints the date and time at which it was run
#

# Output the document header followed by a blank line

echo Content-type: text/plain
echo

# Output the date

echo This document was created on `date`
```

- Generates document
- Document contains three lines of text
 - Header
 - Blank line
 - Document creation date

Long-Term State Information

- Program lifetime
 - CGI program invoked by server
 - Program exits after generating output
- To maintain persistent data
 - Write to file on disk
 - Read from file on disk

Example CGI Script With State Information

```
#!/bin/sh
FILE=ipaddrs

echo Content-type: text/plain
echo

# See if IP address of browser's computer appears in our file

if grep -s $REMOTE_ADDR $FILE >/dev/null 2>&1
then

    echo Computer $REMOTE_ADDR has requested this URL previously.

else

    # Append browser's address to the file

    echo $REMOTE_ADDR >> $FILE
    echo This is the first contact from computer $REMOTE_ADDR

fi
```

- Client's IP address in environment variable
- Check if address in file
- Respond to client

Encoding Information In A URL

- URL can contain arguments
- Question mark separates CGI path from arguments
- Arguments can encode information

Example Of Arguments Encoding Information

```
#!/bin/sh

echo Content-type: text/html
echo

N=$QUERY_STRING
echo "<HTML>"

case "x$N" in

x)      N=1
        echo "This is the initial page.<BR><BR>"
        ;;

x[0-9]*) N=`expr $N + 1`
        echo "You have displayed this page $N times.<BR><BR>"
        ;;

*)      echo "The URL you used is invalid.</HTML>"
        exit 0
        ;;

esac
echo "<A HREF=\"http://$SERVER_NAME$SCRIPT_NAME?$N\">"
echo "Click here to refresh the page.</A> </HTML>"
```

- Argument encodes number of times executed

Example Of Script Execution

- Initial document

```
Content-type: text/html
```

```
<HTML>
```

```
This is the initial page.<BR><BR>
```

```
<A HREF="http://www.nonexist.com/cgi/ex4?1">
```

```
Click here to refresh the page.</A> </HTML>
```

- Resulting display

This is the initial page.

[Click here to refresh the page.](http://www.nonexist.com/cgi/ex4?1)

Example Of Script Execution (continued)

- Generated output

```
Content-type: text/html
```

```
<HTML>
```

```
You have displayed this page 2 times.<BR><BR>
```

```
<A HREF="http://www.nonexist.com/cgi/ex4?2">
```

```
Click here to refresh the page.</A> </HTML>
```

- Resulting display

You have displayed this page 2 times.

[Click here to refresh the page.](http://www.nonexist.com/cgi/ex4?2)

Generated URL Values

When it generates a document, a dynamic document program can embed state information as arguments in URLs. The argument string is passed to the program for the URL, enabling a program to pass state information from one invocation to the next.

Continuously Changing Information

- Needed for
 - Animations
 - Rapid updates (e.g., stock prices)
- Achieved with two mechanisms
 - Server push
 - Active document

Server Push Technology

- Client forms connection
- Server sends updates repeatedly
- Impractical

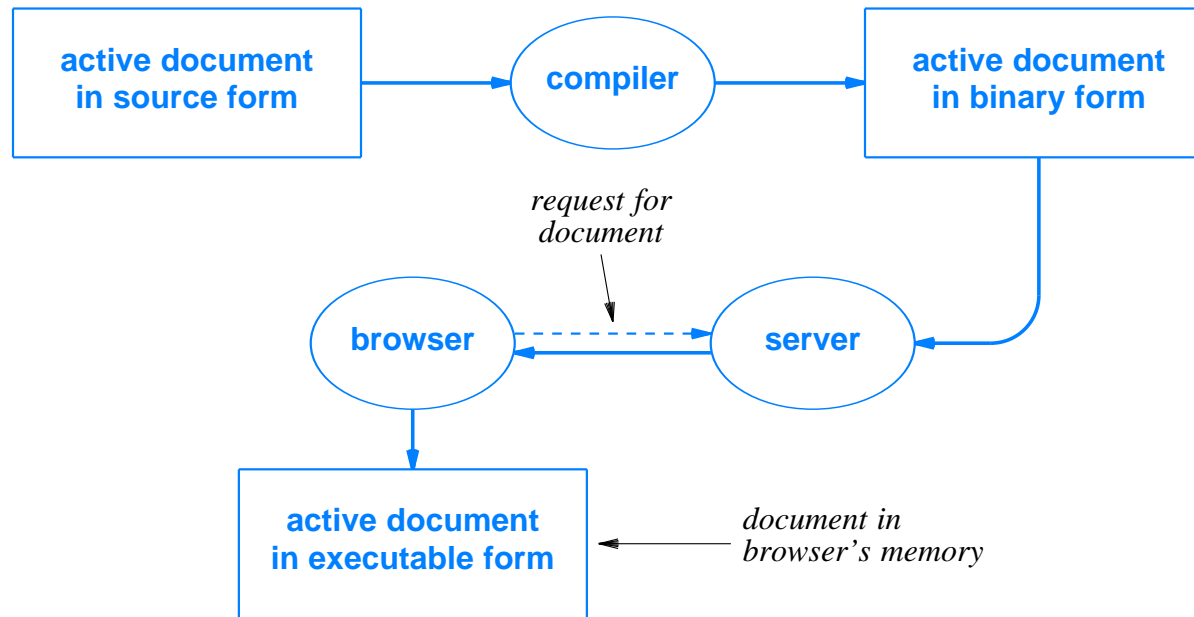
Active Document Technology

- Server
 - Sends computer program to client
- Client
 - Runs program locally
- Program
 - Controls display
 - Interacts with user

Active Document Representation

- Desire
 - Platform independence
 - Efficient execution
 - High-speed data transmission
 - Late binding
- Consequence
 - Compact representation
 - Interpretive execution

Active Document Translation



- Compiler produces machine-independent binary
- Browser interprets binary

Java Technology

- Developed by Sun Microsystems
- Used for
 - Conventional applications
 - Active documents (*applets*)
- Includes
 - Programming language
 - Run-time system
 - Class library

Java Language Characteristics

- High Level
- General Purpose
- Similar to C++
- Object Oriented
- Dynamic
- Strongly typed
- Statically type checked
- Concurrent

Java Run-Time Environment Characteristics

- Interpretative Execution
- Automatic Garbage Collection
- Multi-threaded Execution
- Internet Access
- Graphics Support

Java Library

- Classes for
 - Graphics Manipulation
 - Low-Level Network I/O
 - Interaction With A Web Server
 - Run-Time System Access
 - File I/O
 - Conventional Data Structures
 - Event Capture
 - Exception Handling

Choice Of Graphics Interface

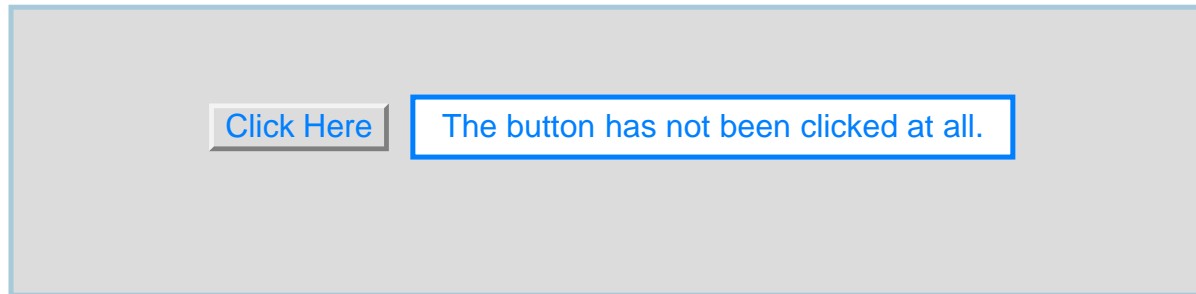
Java includes an extensive graphics toolkit that consists of run-time support for graphics as well as interface software. The toolkit allows a programmer to choose a high-level interface, in which the toolkit handles details, or a low-level interface, in which the applet handles details.

Example Java Applet

- Window with two items
 - Text area
 - Button
- Change text when button clicked

Illustration Of Applet Display

- Initial



- After user clicks button



Example Applet Code

```
import java.applet.*;
import java.awt.*;

public class clickcount extends Applet {
    int count;
    TextField f;

    public void init() {
        count = 0;
        add(new Button("Click Here"));
        f = new TextField("The button has not been clicked at all.");
        f.setEditable(false);
        add(f);
    }

    public boolean action(Event e, Object arg) {
        if (((Button) e.target).getLabel() == "Click Here") {
            count += 1;
            f.setText("The button has been clicked " + count + " times.");
        }
        return true;
    }
}
```


Applet Invocation

- Available in HTML
- Uses *applet* tag
- Specifies
 - *Codebase* (machine and path)
 - *Code* (specific class to run)
- Example

```
<applet codebase="www.nonexist.com/pth"  
        code="bbb.class">
```

Java Functionality

- HTML interface
 - Controls display
 - Interacts with user
- HTTP interface
 - Accesses remote Web documents
 - Invokes other applets
- Exceptions
 - Indicate unanticipated circumstances
 - Can be caught and handled

Middleware

- Tools to help programmers
- Makes client-server programming
 - Easier
 - Faster
- Makes resulting software
 - Less error-prone
 - More reliable

Middleware Approach

- Allow programmer to work with familiar language constructs
- Provide tools to help programmer
 - Special translators
 - Libraries
- Automatically generate code for
 - Network communication
 - Connection management

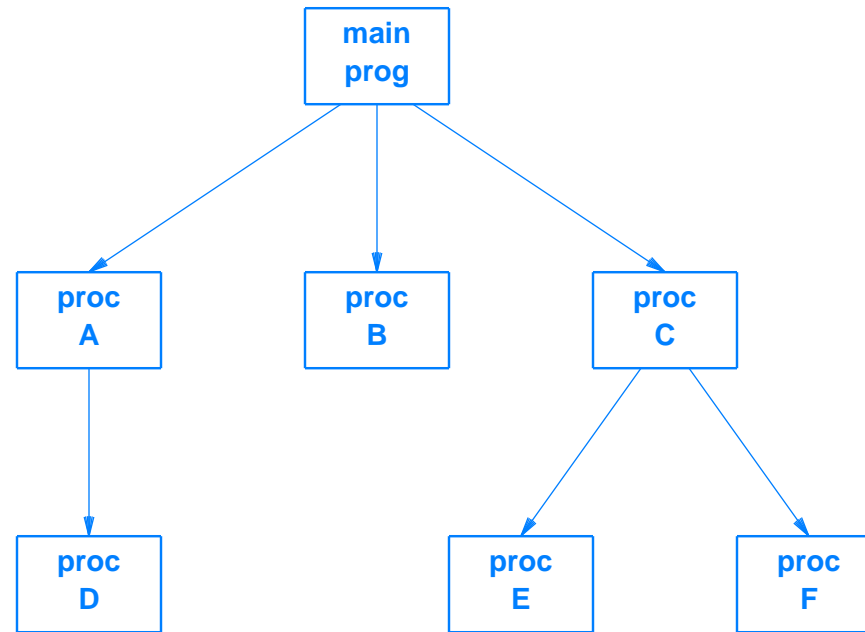
Remote Procedure Call

- Uses standard procedure call paradigm
- Divides program along procedure call boundaries
 - Main program and procedures for user interaction in client side
 - Other procedures in server side

Reason For Remote Procedure Call

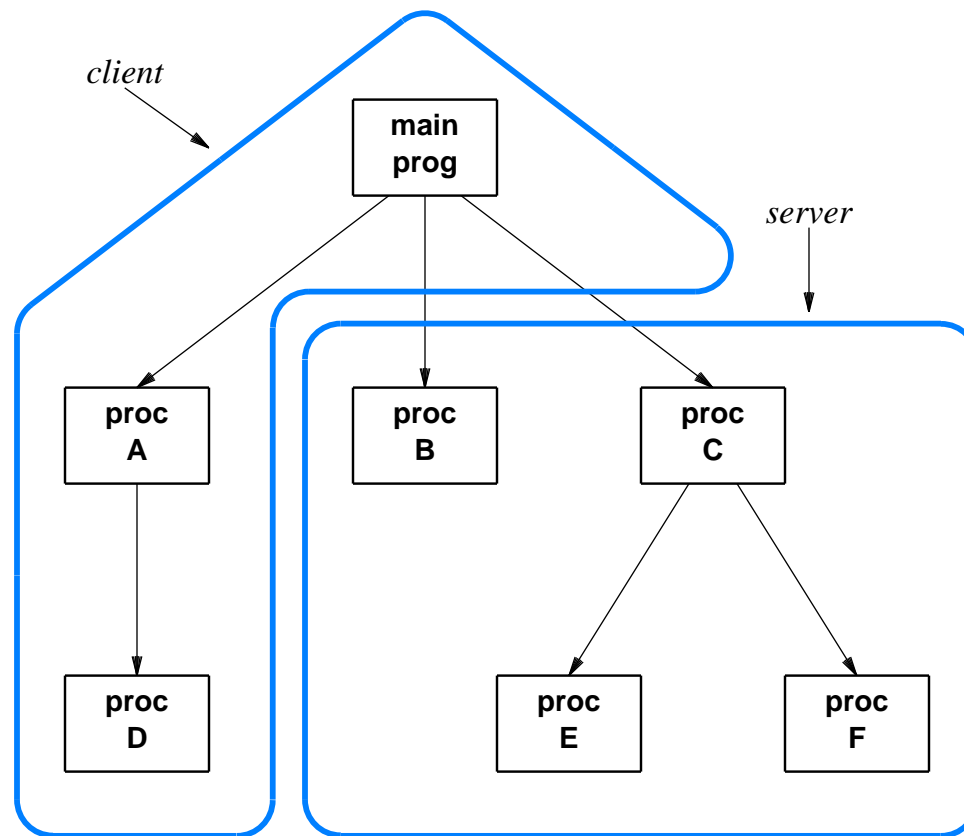
If a programmer follows the same paradigm used to build conventional programs when building client and server software, the programmer will find the task easier and will make fewer mistakes.

Illustration Of Conventional Procedure Call Graph



- Arrow denotes procedure call

Procedure Call Graph Divided Into Client And Server

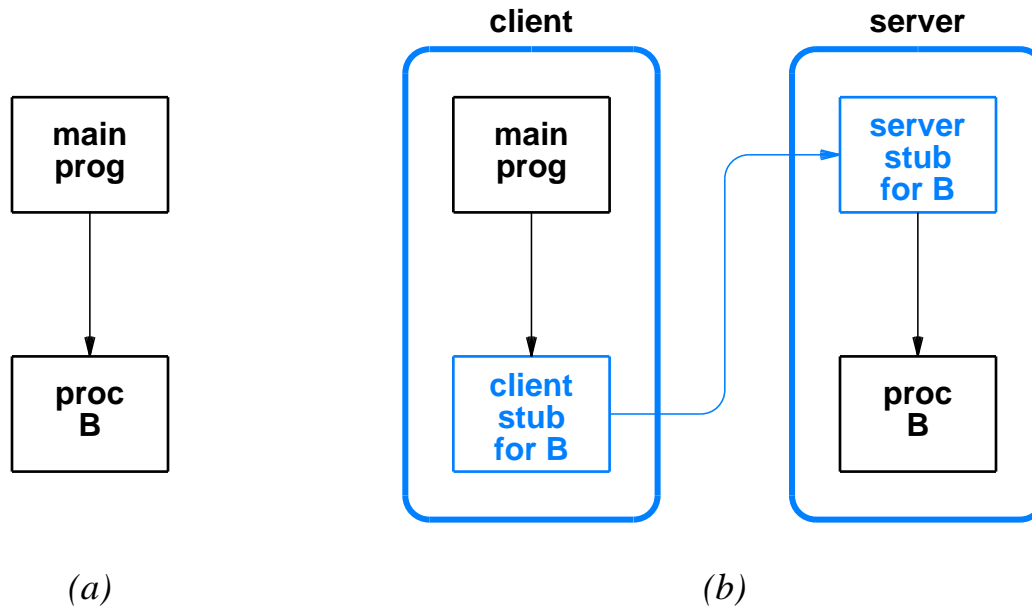


- Division occurs on call boundary
- Main program in client piece

Communication Stubs

- Inserted to enable remote “call”
- Automatically generated
- Use original call interface
- Allow calling and called procedure to remain unchanged

Illustration Of Client And Server Stubs



- Original call in (a)
- Same interface with stubs in (b)

Creating Stubs

- Programmer writes
 - Code for a program
 - Specification of procedure interfaces using *Interface Definition Language (IDL)*
- Middleware generates
 - Client and server stub code
 - Necessary socket calls
 - Data translation

Data Representation

- Network can connect heterogeneous computers
- Two computers may use different
 - Integer representations
 - Character codes
 - Floating point representations
- Translation required

Possible Data Translation Schemes

- Use receiver's representation
 - Sender translates all outgoing data
- Use sender's representation
 - Receiver translates all incoming data
- Use external representation (popular)
 - Sender translates to external form before sending
 - Receiver translates from external form after reception

Middleware Technologies That Use Remote Procedure Call

- ONC RPC
 - Open Network Computing
 - IETF standard
 - Popular in Unix world
- DCE RPC
 - Distributed Computing Environment
 - Open Group Standard

Middleware Technologies That Use Remote Procedure Call (continued)

- MSRPC
 - Microsoft
 - Variant of DCE RPC

Object-Oriented Middleware

- Designed for use with object-oriented programming languages
- Same general scheme as RPC
 - Interface Definition Language
 - Tool to build stubs
 - Libraries to handle network communication
- Uses method invocation instead of procedure call

Middleware Technologies That Use Remote Object Invocation

- CORBA
 - Common Object Request Broker Architecture
 - Best known object-oriented middleware
- MSRPC2
 - Microsoft
 - Also called *Object RPC (ORPC)*

Middleware Technologies That Use Remote Object Invocation (continued)

- COM/DCOM
 - Also from Microsoft
 - Component Object Model (COM)
 - * Used on single computer
 - * Provides mechanism for inter-object references
 - Distributed Component Object Model
 - * Used across multiple computers
 - * Includes communication stubs

Summary

- Web is major application in Internet
- Client
 - Called *browser*
 - Fetches and displays document
- Web documents
 - Stored on servers
 - Standard representation is HTML

Summary (continued)

- HTML
 - Markup language
 - Uses tags embedded in text
- URL components
 - Protocol
 - Domain name of server
 - Protocol port number
 - Path of item
 - Only domain name is required

Summary (continued)

- Static Web page
 - Unchanging
- Dynamic Web page
 - Output from a program on the server
- Active Web page
 - Runs in browser
 - Consists of a computer program

Summary

(continued)

- Dynamic Web page technology
 - Known as CGI
 - CGI program usually a script
 - Document begins with header line
 - URL can contain arguments

Summary (continued)

- Active Web page technology
 - Known as Java
 - Programming language plus runtime support
 - Document called *applet*

Summary (continued)

- Middleware
 - Tools to help build client and server
 - Automates routine tasks
 - Two popular paradigms
 - * Remote procedure call
 - * Object invocation
 - Generates communication stubs