CS342: Organization of Prog. Languages

Topic 2: Language Examples

- Assembly Language
- C
- Fortran
- Cobol
- Java
- Common Lisp
- Yacc
- XSLT
- VHDL
- Icon
- DNS Configuration

© 2000-2007 Stephen M. Watt.

Example: Assembly Language

```
        ...
        ...
        pushl $.LC0
.LCFI10:
        call getenv
        movl %eax,-44(%ebp)
        addl $20,%esp
.LCFI11:
        movl %eax,%ebx
        shrl $14,%ebx
        movl -28(%ebp),%eax
        shrl $14,%eax
        subl %eax,%ebx
        movl %ebx,%edx
        movl -24(%ebp),%eax
        movb (%ebx,%eax),%al
        andb $31,%al
        andl $255,%eax
        cmpl $3,%eax
        jbe .L128
        leal -6(,%eax,2),%ecx
        movl $1,%eax
        sall %cl,%eax
.L128:
        movl -24(%ebp),%esi
        .align 4
```

```
.L132:
        cmpl $3,%eax
        jbe .L134
        subl %eax,%edx
        movb (%edx,%esi),%al
        andb $31,%al
        andl $255,%eax
        cmpl $3,%eax
        jbe .L132
        leal -6(,%eax,2),%ecx
        movl $1,%eax
        sall %cl,%eax
        jmp .L132
        .align 4
.L134:
        pushl %edx
.LCFI12:
        pushl %ebx
.LCFI13:
        pushl $.LC1
.LCFI14:
        call printf
        addl $12,%esp
        ...
```

```
        ...
        ...
        .byte        0xe
        .byte        0x8
        .byte        0x85
        .byte        0x2
        .byte        0x4
        .4byte        .LCFI1-.LCFI0
        .byte        0xd
        .byte        0x5
        .byte        0x4
        .4byte        .LCFI3-.LCFI1
        .byte        0x87
        .byte        0xe
        .byte        0x4
        .4byte        .LCFI4-.LCFI3
        .byte        0x86
        .byte        0xf
        .byte        0x4
        .4byte        .LCFI5-.LCFI4
        .byte        0x83
        .byte        0x10
        .byte        0x4
        ...
        ...
        ...
```

Example: C

```
main()
{
    int     i, j;

    printf("|");
    for (j = 0 ; j <= 0xFF; j++) {
        if (j != 0 && j % 32 == 0) printf("|\n|");

        if (j < 0x20 || j == 0x7F || 0x80 <= j && j < 0xA0)
            i = ' ';
        else
            i = j;

        printf("%c", i);
    }
    printf("|\n");
}
```

Example: Java

```java
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.util.*;


public class LnFListener
   implements ActionListener
{

   private Frame frame;

   public LnFListener(Frame f) { frame = f; }


   public void actionPerformed(ActionEvent e) {

      String lnfName =
         "javax.swing.plaf.metal.MetalLookAndFeel";
      String cmd =
         e.getActionCommand();

      UIManager.LookAndFeelInfo uLF[] =
         UIManager.getInstalledLookAndFeels();
```

```java
for (int i = 0; i < uLF.length; i++) {
   if (uLF[i].getName() == cmd) {
      lnfName = uLF[i].getClassName();
      break;
   }
}
try {
   UIManager.setLookAndFeel(lnfName);
   System.out.println("Setting look & feel to "
                      + lnfName);
   SwingUtilities.updateComponentTreeUI(frame);
}
catch (InstantiationException e3) {
   ZError.error("Could not load LookAndFeel: "
                + lnfName);
}
catch (IllegalAccessException e4) {
   ZError.error("Cannot use LookAndFeel: "
                + lnfName);
}
   }
}
```

Example: Common Lisp

```lisp
;;; Compile a "foo" file to lisp, then object code.
;;; Then load it into the current work-space.

(defun compile-foo-file (file &optional (opts nil))
  (let* ((path  (pathname file))
         (name  (pathname-name path))
         (dir   (pathname-directory path))
         (type  (pathname-type path))
         (lpath (make-pathname :name name :type "l"))
         (cpath (make-pathname :name name :type "o")) )

    ; If no file type then use "foo"
    (if (null type)
        (setq path (make-pathname :directory dir
                                  :name name :type "foo")) )

    ; Compile foo file to a lisp file.
    (if opts
        (system (format nil "compfoo ~A -Flsp ~A" opts (namestring path)))
        (system (format nil "compfoo -Flsp ~A" (namestring path))) )

    ; Compile then load resulting lisp file.
    (compile-file (namestring lpath))
    (load (namestring cpath)) ))
```

Example: Fortran

```
      DIMENSION X(100), Y(100), Z(100)
      N=100
      DO 10 I=1,N
         X(I)=I*(1.0/N)
 10   CONTINUE
      CALL PYTHAG(N,X,Y,Z)
      DO 20 J=1,5
         PRINT 90, X(J), Y(J), Z(J)
 20   CONTINUE
      STOP
 90   FORMAT(2X,3F10.3)
      END
C
      SUBROUTINE PYTHAG(N,X,Y,Z)
      DIMENSION X(1), Y(1), Z(1)
      DO 10 I=1,N
         Z(I) = SQRT(X(I)*X(I) + Y(I)*Y(I))
         ENDIF
 10   CONTINUE
      RETURN
      END
```

Example: Cobol

Sample Cobol program from the University of Limerick. (www.csis.ul.ie/COBOL)

```
      $ SET SOURCEFORMAT"FREE"
IDENTIFICATION DIVISION.
PROGRAM-ID. Validate IS INITIAL.
AUTHOR.  Michael Coughlan.

DATA DIVISION.
WORKING-STORAGE SECTION.
01 MonthDayTable.
   02 TableValues           PIC X(24)
            VALUE "312831303130313130313031".
   02 FILLER REDEFINES TableValues.
      03 DaysInMonth
            OCCURS 12 TIMES PIC 99.


01 CurruptDate              PIC 9(8).

01 LeapQuot                 PIC 9(4).
01 LeapRemain               PIC 9(4).

01 FILLER                   PIC 9 VALUE ZERO.
   88 LeapYear              VALUE 1.
```

```
LINKAGE SECTION.
01 InputDateLA.
   02 DayLA                 PIC 99.
   02 MonthLA               PIC 99.
      88 MonthInvalid       VALUE 13 THRU 99.
      88 MonthIsFebruary    VALUE 2.
   02 YearLA                PIC 9(4).

01 ValidationResultLB       PIC 9.
   88 DateIsValid           VALUE 0.
   88 DateNotNumeric        VALUE 1.
   88 YearContainsZeros     VALUE 2.
   88 MonthContainsZeros    VALUE 3.
   88 DayContainsZeros      VALUE 4.
   88 MonthGreaterThan12    VALUE 5.
   88 DayTooGreatForMonth   VALUE 6.

PROCEDURE DIVISION USING InputDateLA, ValidationResultLB.
Begin.
   EVALUATE TRUE
     WHEN InputDateLA NOT NUMERIC  SET DateNotNumeric     TO TRUE
     WHEN YearLA EQUAL TO ZEROS    SET YearContainsZeros  TO TRUE
     WHEN MonthLA EQUAL TO ZEROS   SET MonthContainsZeros TO TRUE
     WHEN DayLA EQUAL TO ZEROS     SET DayContainsZeros   TO TRUE
     WHEN MonthInvalid             SET MonthGreaterThan12 TO TRUE
     WHEN OTHER PERFORM CheckForValidDay
   END-EVALUATE

   EXIT PROGRAM.
```

```
CheckForValidDay.
*  Years evenly divisible by 4 are leap years, but
*  years evenly divisible by 100 are not leap years, but
*  years evenly divisible by 400 are leap years.

   DIVIDE YearLA BY 400 GIVING LeapQuot REMAINDER LeapRemain.
   IF LeapRemain = 0
      SET LeapYear TO TRUE
    ELSE
      DIVIDE YearLA BY 100 GIVING LeapQuot REMAINDER LeapRemain
      IF LeapRemain NOT = 0
         DIVIDE YearLA BY 4 GIVING LeapQuot REMAINDER LeapRemain
         IF LeapRemain = 0
            SET LeapYear TO TRUE
         END-IF
      END-IF
   END-IF

  IF LeapYear AND MonthIsFebruary
         MOVE 29 TO DaysInMonth(2)
  END-IF
  IF DayLA GREATER THAN DaysInMonth(MonthLA)
     SET DayTooGreatForMonth TO TRUE
   ELSE
     SET DateIsValid TO TRUE
  END-IF.
```

Example: YACC

```
%%
ExternalDeclaration
: FunctionDefinition { yytree = $$ = $1; YYACCEPT; }
| Declaration        { yytree = $$ = $1; YYACCEPT; }
| CTOK_EOF           { yytree = $$ = $1; YYACCEPT; }
;

FunctionDefinition
: Declarator optseq(Declaration) CompoundStatement
        { $$ = ccNewFDef(0,  0, $1, $2, $3); }
| seq(DeclarationSpecifier)
  Declarator optseq(Declaration) CompoundStatement
        { $$ = ccNewFDef(0, $1, $2, $3, $4); }
;

Declaration
: seq(DeclarationSpecifier) optlist(InitDeclarator) CTOK_Semi
        { $$ = ccDoTypedefs(ccNewDecl(0, $1, $2, $3)); }
;

DeclarationSpecifier
: StorageClassSpecifier
| TypeSpecifier
| TypeQualifier
;
```

Example: XSLT

```
<!-- ROOT -->

<xsl:template match = "/">
  <xsl:apply-templates mode = "root"/>
</xsl:template>

<xsl:template match = "*" mode = "root">
  <math>
    <semantics>
      <math>
        <xsl:apply-templates mode = "semantics"/>
      </math>
      <annotation-xml encoding="MathML">
        <math>
        <xsl:copy-of select = "./*"/>
        </math>
      </annotation-xml>
    </semantics>
  </math>
</xsl:template>
```

```
<!-- SEMANTICS CONTAINERS -->

<xsl:template match = "*" mode = "semantics">
<semantics>
  <xsl:choose>
    <xsl:when test="self::semantics">
      <xsl:apply-templates select="*[1]"/>
      <xsl:copy-of select="annotation-xml"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:apply-templates select="."/>
      <annotation-xml encoding="MathML">
      <xsl:copy-of select="."/>
      </annotation-xml>
    </xsl:otherwise>
  </xsl:choose>
</semantics>
</xsl:template>

<xsl:template match = "semantics">
  <xsl:apply-templates select="*[1]" mode = "semantics"/>
</xsl:template>
```

```
<!-- BASIC ELEMENTS -->

<xsl:template match = "cn">
  <mn>
    <xsl:apply-templates mode = "semantics"/>
  </mn>
</xsl:template>

<xsl:template match = "ci">
  <mi>
    <xsl:apply-templates mode = "semantics"/>
  </mi>
</xsl:template>
```

Example: VHDL

```
ENTITY moore_110_detector IS PORT (x, clk : IN BIT; z : OUT BIT);
END moore_110_detector;
--
ARCHITECTURE behavioral OF moore_110_detector IS
TYPE state IS (reset, goto1, goto11, goto110);
SIGNAL current : state := reset;
BEGIN
  PROCESS(clk)
  BEGIN
    IF clk = '1' THEN
      CASE current IS
      WHEN reset =>
        IF x = '1' THEN current <= goto1;
        ELSE current <= reset; END IF;
      WHEN goto1 =>
        IF x = '1' THEN current <= goto11;
        ELSE current <= reset; END IF;
      WHEN goto11 =>
        IF x = '1' THEN current <= goto11;
        ELSE current <= goto110; END IF;
      WHEN goto110 =>
        IF x = '1' THEN current <= goto1;
        ELSE current <= reset; END IF;
      END CASE;
    END IF;
  END PROCESS;
  z <='1' WHEN current = goto110 ELSE '0';
END behavioral;
```

Example: Icon

```
# From "An Overview of the Icon Programming Language; Version 9",
# by Ralph E. Griswold

global uses, lineno, width

procedure main(args)
   width := 15                # width of word field
   uses := table()
   lineno := 0
   every tabulate(words())    # tabulate all citations
   output()                   # print the citations
end

#  Add line number to citations for word
#
procedure tabulate(word)
   /uses[word] := set()
   insert(uses[word], lineno)
   return
end
```

```
#  Generate words
#
procedure words()
   while line := read() do {
      lineno +:= 1
      write(right(lineno, 6), "  ", line)
      map(line) ? while tab(upto(&letters)) do {
         s := tab(many(&letters))
         if *s >= 3 then suspend s# skip short words
         }
      }
end

#  Print the results
#
procedure output()
   write()                   # blank line
   uses := sort(uses, 3)     # sort citations
   while word := get(uses) do {
      line := ""
      numbers := sort(get(uses))
      while line ||:= get(numbers) || ", "
      write(left(word, width), line[1:-2])
      }
end
```

Example: DNS configuration file

```
; Zone file for funkydom.ca
@    IN    SOA    ns1.funkydom.ca. root.ns1.funkydom.ca. (
            2000090900    ; unique serial number YYYYMMDDNN
            8H        ; refresh time
            2H        ; retry time
            1W        ; expire time
            1D )        ; minimum
;
        TXT     "FunkyDom.Ca,  Your source for Funky code"
        NS      ns1            ; name server name
        MX      10 ns1.funkydom.ca.    ; primary mail exchanger
;
localhost  A    127.0.0.1
;
gw      A       192.168.9.1
        HINFO   "Pentium 90" "Redhat Linux 5.2"
        TXT     "Gateway computer"
www     CNAME   gw


ns1     A      192.168.9.2
        HINFO  "AMD K6 233"   "Redhat Linux 6.2"
        TXT    "Local network server"
```

```
devel   A       192.168.9.3
        HINFO  "Pentium 233"  "Windows 2000"
        TXT    "Development machine"

research A      192.168.9.5
        HINFO  "AMD Athalon 1000" "Redhat Linux 6.2"
        TXT    "SMW's UWO research machine running Linux"

tester  A      192.168.9.6
        HINFO  "Pentium 450"  "Windows 98"
        TXT    "Testing machine"
```