```
Pass 2.1
 *** Compilers 31/7/76


   static
   §s
       RtDf    = false
 5     CaseNo  = 0
   $s

   let Pass2[] be
   §P2
10     Send := Write
       SpecSend := Write
       ReportNo := 0
       PassNo := 2
       UpdateCh[DUMMY]
15     RtDf := false
       Send[ENDPROG]
       BackPass[]
       CalculateStructureSize[]
   $P2
20
   and BackPass[] be
   §BP
       let Assignment = false
       and Definition = false
25     and AssNo = 1
       and RCaseNo = 0
       §R  Scan[]
           switchon ChType into
           §S
30             case COLON:
                       if Assignment do
                       §   Send[AssNo ∨ COUNT]
                           Send[LASS]
                           Assignment := false
35                     $
                       Read[]
                       unless IsName[Ch] do
                           Error[201, HARD, Ch, DeleteCommand]
                       FindLabel[]
40                     Insert[LABEL]
                       endcase

               case CASE:
                       CaseNo := CaseNo+1
45                     endcase

               case ENDSWITCH:
                       RCaseNo := CaseNo
                       CaseNo := 0
50                     BackPass[] repeatuntil Ch = SWITCHON
                       Send[CaseNo ∨ COUNT]
                       CaseNo := RCaseNo
                       endcase

55             case FNDF:
                       if RtDf do Ch := RTDF
                       RtDf := false
                       endcase

60             case MANFNDF:
                       if RtDf do Ch := MANRTDF
```

```
                RtDf := false
                endcase

    case ASS:
                AssNo := 1
                Assignment := true
                endcase

    case DEF:
                Definition := true
                endcase

    case COMMA:
                AssNo := AssNo + 1
                endcase

    case COND:
                AssNo := AssNo-1
                endcase

    case REPEAT:
    case REPEATUNTIL:
    case REPEATWHILE:
                BackPass[]
                Send[REPSTART]
                return

    case RTBODY:
                RtDf := true
    case DO:
    case OR:
    case IFSO:
    case IFNOT:
    case VALOF:
    case SEMICOLON:
    case CCOLON:
    case SECTBRA:
                if Assignment do
                §   Send[AssNo ∨ COUNT]
                    Send[LASS]
                §
                return

    case SWITCHON:
    case SBRA:
    case VECAP:
                return

    case SKET:
    case ENDVALOF:
                BackPass[]
                endcase

    case LET:
    case AND:
                if Definition do
                §   Send[LOCDEF]
                    Definition := false
                §
                endcase

    case SECTKET:
                BackPass[] repeatuntil IsSecBra[Ch]
                endcase
```

```
                case ENDPROG:
                        return

130             case ERROR:
                        DeleteCommand[]
                        BackUp[DUMMY]
                case default:
                        endcase
135
            $S
        $R repeat
    $BP

140 and FindLabel[] be
    §FL
        let OldNameBlock = NameBlock[Ch]
        unless LevelType[PresentLevel] = LABEL do
        §   let L = FormLevel[PreviousLevel[PresentLevel],
145                           LevelType[PresentLevel],
                              Generation[PresentLevel],
                              NameList[PresentLevel]]
            UpdatePreviousLevel[PresentLevel, L]
            UpdateNameList[PresentLevel, NILNAME]
150         UpdateLevelType[PresentLevel, LABEL]
            UpdateGeneration[PresentLevel, Generation[PresentLevel] + 1]
        §
        if Generation[PresentLevel] = Generation[OldNameBlock] do
            Error[202, HARD, Ch, NullProgram]
155     AddName[Ch, LABEL]
    §FL

    and IsSecBra[C] = IsCType[C, SECTBRA]

160 and CalculateStructureSize[] be
        ReportS['Structure Size = *N',
            StructureSize - StructureNPtr + StructureLPtr]

    and DeleteCommand[] be
165 §DC
        Read[]
        switchon ChType into
        §S
            case SEMICOLON:
170         case RTBODY:
            case DO:
            case OR:
            case IFSO:
            case IFNOT:
175         case VALOF:
            case CCOLON:
            case SECTBRA:
            case ENDBODY:
                    return
180
            case ENDPROG:
                    CompilerError[2101]

            case SECTKET:
185                 DeleteUntil[SECTBRA]
                    endcase

            case ENDVALOF:
                    DeleteUntil[VALOF]
190                 endcase
        $S
```

```
    §DC repeat

    and DeleteUntil[x] be
195 §DU
        Read[]
        switchon ChType into
        §S
            case SECTKET:
200                 DeleteUntil[SECTBRA]
                    endcase

            case ENDVALOF:
                    DeleteUntil[VALOF]
205                 endcase

            case ENDPROG:
                    CompilerError[2102]

210         default:if ChType = x return
        §S
    §DU repeat

****
```