

Pass 4.1

\*\*\* Compilers 3/7/76

```

    let Pass4[] be
    §P4
        SetupPass4[]
5        Peep[1]
        NewSection[]
        Code[]
        Data[]
        Labels[]
10       Globals[]
        Diagnostics[]
        EndLoad[]
        ClosePass4[]
        Peep[3]
15 §P4

    and NewSection[] be
    §NS
20     Ptr := 0
        OuttoBody[STACKMACHINECODE]
        OutputSection[NEWSECTION, Ptr]
    §NS

25
    and Code[] be
    §C
        SetUpCode[]
        GenerateCode[]
30     Relocate[]
        AlterBody[]
        OutputSection[CODE, Ptr]
        CSize := Ptr
        ResetBody[]
35 §C

    and Data[] be
    §D
40     Ptr := 0
        for i = 1 to SN do
            unless StaticType[i] = CSEG do
                OuttoBody[StaticType[i] = LIT → StaticVal[i], UNSET]
        for i = 1 to SN do
45         if StaticType[i] = CSEG do
            OuttoBody[UNSET]
        CopytoBody[TableData, TableMax]
        TransferInStrings[]
        unless Ptr = 0 do OutputSection[DATA, Ptr]
50     ReportS['C, D = *N, *N', CSize, Ptr]
    §D

    and Labels[] be
55 §B
        let j = 0
        Ptr := 0
        for i = 1 to SN do
            unless StaticType[i] = CSEG do
60         § if StaticType[i] = DSEG do
                Pair[DSEG, DSEG, StaticVal[i], j]
```

```

        unless StaticLastUse[i] = UNSET do
            Pair[DSEG, CSEG, j, StaticLastUse[i]]
            j := j + 1
65    $
    for i = 1 to SN do
        if StaticType[i] = CSEG do
            $ Pair[CSEG, DSEG, StaticVal[i], j]
            unless StaticLastUse[i] = UNSET do
70                Pair[DSEG, CSEG, j, StaticLastUse[i]]
                j := j + 1
            $
        for i = 1 to MFN do
            unless MFLastUse[i] = UNSET do
75                Pair[CSEG, CSEG, MFVal[i], MFLastUse[i]]
            for i=1 to SL do
                unless StringUse[i] = UNSET do
                    Pair[DSEG, CSEG, StringData[i], StringUse[i]]
                OutputSection[LABELS, Ptr]
80    unless (Mode ^ 1) = 0 do
        ReportS['Labels Size = *N',Ptr]
    $B

85 and Globals[] be
    $G
        let RepeatedGlob = false
        Ptr := 0
        for i = 1 to GN do
90    $ for j = i + 1 to GN do
            if GlobalLoc[i] = GlobalLoc[j] do
                $ Error[22, HARD, GlobalLoc[i], NullProgram]
                RepeatedGlob := true
                break
95    $
            OuttoBody[GlobalLoc[i]]
            OuttoBody[GlobalType[i]]
            OuttoBody[GlobalVal[i]]
        $
100    if RepeatedGlob do Finish[]
        OutputSection[GLOBALS, Ptr]
        unless (Mode ^ 1) = 0 do
            ReportS['Globals Size = *N',Ptr]
    $G
105

    and Diagnostics[] be
    $D
        let S = InfromFile[WorkFile['Names']]
110    let TotalNames = Next[S]
        for n = 1 to TotalNames do
            $f
                let N = ReadString[S]
                and Used = false
115    for i = 1 to DN do
                if DiagName[i] = n do
                    $ DiagString[i] := N
                    Used := true
                $
120    unless Used do ReturnString[N]
    $f
        Close[S]

        Ptr := 0
125    OuttoBody[DN]
        for i = DN to 1 by -1 do

```

```

        $f
        OuttoBody[DiagAddr↓i]
        OuttoBody[DiagGlobNo↓i]
130      CopyStringtoBody[DiagString↓i]
        $f
        OutputSection[DIAGNOSTICS, Ptr]
        unless (Mode ^ 1) = 0 do
            ReportS['Diagnostics size = *N', Ptr]
135 $D

        and EndLoad[] be
        $EL
140      OutputSection[ENDLOAD, 0]
        $EL

        and SetupPass4[] be
145 $SCG
        PassNo := 4
        ReportNo := 0
        if SN + TableMax > CodeSize do
            Error[10, HARD, DUMMY, Finish]
150      GlobalType := ProperVec[GN]
            GlobalLoc := ProperVec[GN]
            GlobalVal := ProperVec[GN]
            StaticType := ProperVec[SN]
            StaticLastUse := ProperVec[SN]
155      StaticVal := ProperVec[SN]
            MFVal := ProperVec[MFN]
            MFLastUse := ProperVec[MFN]
            SetupStrings[]
            StringUse := ProperVec[SL]
160      StringData := ProperVec[SL]
            TableData := ProperVec[TableMax]
            TDPtr := TableMax
            HopList := ProperVec[HN]
            BackHopList := ProperVec[HN]
165      SwitchList := ProperVec[SWN]
            DiagAddr := ProperVec[GN + SN]
            DiagGlobNo := ProperVec[GN + SN]
            DiagName := ProperVec[GN + SN]
            DiagString := ProperVec[GN + SN]
170      DN := 0
            SetUpBody[CodeSize]
            for i=1 to GN do GlobalType↓i := UNSET
            for i=1 to SN do StaticType↓i := UNSET
            for i=1 to SN do StaticLastUse↓i := UNSET
175      for i = 1 to MFN do MFLastUse↓i := UNSET
            for i=1 to SL do StringUse↓i := UNSET
            for i=1 to HN do HopList↓i := UNSET
        $SCG

180
        and SetupStrings[] be
        $SS
            NIn := InfromFile[WorkFile['Strings']]
            SL := Next[NIn]
185 $SS

        and TransferInStrings[] be
        $TIS
190      for i = 1 to SL do
            $ let x = Next[NIn]

```

```

        let l = (x rshift 8) / 2
        StringData[i] := Ptr
        if Ptr + l ≥ CodeSize do Error[10, HARD, DUMMY, Finish]
195      UpdateBody[Ptr, x]
        TransferIntoBody[NIn, Ptr + 1, l]
        unless StringUse[i] = UNSET do Ptr := Ptr + l + 1
    $
    Close[NIn]
200 $TIS

```

```

    and ClosePass4[] be
    $CCG
205      ReturnVec[GlobalType, GN]
        ReturnVec[GlobalLoc, GN]
        ReturnVec[GlobalVal, GN]
        ReturnVec[StaticType, SN]
        ReturnVec[StaticLastUse, SN]
210      ReturnVec[StaticVal, SN]
        ReturnVec[MFVal, MFN]
        ReturnVec[MFLastUse, MFN]
        ReturnVec[StringUse, SL]
        ReturnVec[StringData, SL]
215      ReturnVec[TableData, TableMax]
        ReturnVec[HopList, HN]
        ReturnVec[BackHopList, HN]
        ReturnVec[SwitchList, SWN]
        ReturnVec[DiagAddr, GN + SN]
220      ReturnVec[DiagGlobNo, GN + SN]
        ReturnVec[DiagName, GN + SN]
        ReturnVec[DiagString, GN + SN]
        ReturnBody[CodeSize]
        unless (Mode ∧ 1) = 0 do
225          ReportS['*N hops', HN]
    $CCG

```

```

    and ReadString[S] = valof
230 $RS
        let x = Next[S]
        let l = LHalf[x]/2
        let v = ProperVec[l]
        v↓0 := x
235      TransferIn[S, v + 1, l]
        resultis v
    $RS

```

```

240 and ReturnString[N] be
    ReturnVec[N, LHalf[N↓0]/2]

```

```

    and ProperVec[n] = valof
245 $PV
        let m = MaxVecSize[]
        unless n ≤ m do
            $ Peep[2]
                Error[9, HARD, DUMMY, Finish]
250      $
        $ let V = NewVec[n]
            V↓0 := n
            resultis V
    $PV
255

```

```

and OuttoBody[C] be
$OV
    if Ptr > CodeSize do
260     Error[7,HARD,DUMMY,Finish]
        UpdateBody[Ptr, C]
        Ptr := Ptr + 1
    $OV

265
and Pair[ASeg, PSeg, AEntry, PEntry] be
$P
    OuttoBody[AEntry ∨ ASeg]
    OuttoBody[PEntry ∨ PSeg]
270 $P

and Relocate[] be
$R
275     let Shift = FirstCode
        for i = 1 to SN do
            if StaticType↓i = CSEG do
                StaticVal↓i := StaticVal↓i - Shift
            for i = 1 to GN do
280             if GlobalType↓i = CSEGg do
                GlobalVal↓i := GlobalVal↓i - Shift
            for i = 1 to MFN do
                MFVal↓i := MFVal↓i - Shift
            for i = 1 to DN do
285             DiagAddr↓i := DiagAddr↓i - Shift
            for i = 1 to SN do RelocateChain[lv StaticLastUse↓i, Shift]
            for i = 1 to MFN do RelocateChain[lv MFLastUse↓i, Shift]
            for i = 1 to SL do RelocateChain[lv StringUse↓i, Shift]
        $R
290

and RelocateChain[Lv, Shift] be
unless rv Lv = UNSET do
$u
295     let Loc = rv Lv
        rv Lv := Loc - Shift
        until CodeWord[Loc] = UNSET do
            § let TempLoc = CodeWord[Loc]
                UpdateCode[Loc, TempLoc - Shift]
300         Loc := TempLoc
    $
$u

```

\*\*\*\*