```
-- MiniNub.Mesa
-- Edited by Sandman on October 6, 1977  4:02 PM

DIRECTORY
  AltoFileDefs: FROM "altofiledefs",
  ControlDefs: FROM "controldefs",
  CoreSwapDefs: FROM "coreswapdefs",
  FrameDefs: FROM "framedefs",
  ImageDefs: FROM "imagedefs",
  IODefs: FROM "iodefs",
  LoaderDefs: FROM "loaderdefs",
  SegmentDefs: FROM "segmentdefs",
  StreamDefs: FROM "streamdefs",
  StringDefs: FROM "stringdefs";

MiniNub: PROGRAM
  IMPORTS CoreSwapDefs, LoaderDefs, SegmentDefs, StreamDefs, StringDefs =

BEGIN

Program: TYPE = PROGRAM;

CallDebugger: PROCEDURE =
  BEGIN -- user's entry point to debugger
  state: ControlDefs.StateVector;
  state.stkptr ← 0;
  state.X ← defaultframe;
  CoreSwapDefs.CoreSwap[explicitcall, @state];
  RETURN
  END;

BadFile: SIGNAL [name: STRING] = CODE;

LoadNew: PROCEDURE [name: STRING] RETURNS [Program] =
  BEGIN
  g: ControlDefs.GlobalFrameHandle;
  g ← FrameDefs.New[name ! BadFile, LoaderDefs.VersionMismatch, UNWIND => NULL; ANY => ERROR BadFile[na
**me]];
  IF g # ControlDefs.NULLFrame THEN defaultframe ← g;
  RETURN[LOOPHOLE[g]]
  END;

defaultframe: ControlDefs.GlobalFrameHandle ← REGISTER[ControlDefs.Greg];

comcmRequest: short ImageDefs.FileRequest ← [
  body: short[fill:, name: "Com.Cm."],
  file: NIL,
  access: SegmentDefs.Read,
  link: ];

Done: SIGNAL = CODE;

debug, start: BOOLEAN;

ProcessSwitches: PROCEDURE [s: STRING] =
  BEGIN
  i: CARDINAL;
  inverse: BOOLEAN ← FALSE;
  FOR i IN [0..s.length) DO
    SELECT s[i] FROM
      'd, 'D => inverse ← ~(debug ← TRUE);
      's, 'S => IF inverse THEN inverse ← start ← FALSE ELSE start ← TRUE;
      '- => inverse ← TRUE;
      ENDCASE => inverse ← FALSE;
    ENDLOOP;
  END;

InitSwitches: PROCEDURE = BEGIN debug ← FALSE; start ← TRUE; END;

GetToken: PROCEDURE [com: StreamDefs.StreamHandle, token, ext, switches: STRING] =
  BEGIN
  s: STRING;
  c: CHARACTER;
  token.length ← ext.length ← switches.length ← 0;
  s ← token;
  WHILE ~com.endof[com] DO
```

```
    SELECT (c ← com.get[com]) FROM
      IODefs.SP, IODefs.CR =>
        IF token.length # 0 OR ext.length # 0 OR switches.length # 0 THEN RETURN;
      '. => s ← ext;
      '/ => s ← switches;
      ENDCASE => StringDefs.AppendChar[s, c];
    ENDLOOP;
  RETURN
  END;

LoadSystems: PROCEDURE =
  BEGIN
  user: PROGRAM;
  fa: AltoFileDefs.FA;
  IF comcmRequest.file = NIL THEN RETURN;
  SegmentDefs.LockFile[comcmRequest.file];
  InitSwitches[];
  SkipImage[@fa];
  IF debug THEN CallDebugger[];
  DO
    InitSwitches[];
    user ← LoadUser[@fa ! Done => EXIT];
    IF debug THEN CallDebugger[];
    IF start AND user # LOOPHOLE[ControlDefs.NULLFrame] THEN START user;
    ENDLOOP;
  SegmentDefs.UnlockFile[comcmRequest.file];
  SegmentDefs.ReleaseFile[comcmRequest.file];
  END;

LoadUser: PROCEDURE [fa: POINTER TO AltoFileDefs.FA] RETURNS [user: Program] =
  BEGIN OPEN IODefs, StreamDefs;
  com: StreamHandle;
  name: STRING ← [40];
  ext: STRING ← [10];
  switches: STRING ← [5];
  com ← CreateByteStream[comcmRequest.file, Read];
  BEGIN
  StreamDefs.JumpToFA[com, fa ! ANY => GO TO finished];
  GetToken[com, name, ext, switches];
  IF name.length = 0 THEN GO TO finished;
  IF ext.length = 0 THEN ext ← "bcd";
  StringDefs.AppendChar[name, '.];
  StringDefs.AppendString[name, ext];
  StreamDefs.GetFA[com, fa];
  com.destroy[com];
  user ← LoadNew[name];
  ProcessSwitches[switches];
  EXITS
    finished => BEGIN com.destroy[com]; SIGNAL Done; END;
  END;
  END;

SkipImage: PROCEDURE [fa: POINTER TO AltoFileDefs.FA] =
  BEGIN OPEN IODefs, StreamDefs;
  com: StreamHandle;
  name: STRING ← [40];
  ext: STRING ← [10];
  switches: STRING ← [5];
  com ← CreateByteStream[comcmRequest.file, Read];
  StreamDefs.GetFA[com, fa];
  GetToken[com, name, ext, switches];
  IF StringDefs.EquivalentString[ext, "image"] THEN StreamDefs.GetFA[com, fa];
  com.destroy[com];
  ProcessSwitches[switches];
  END;

-- Main body

ImageDefs.AddFileRequest[@comcmRequest];

[] ← StreamDefs.GetDefaultKey[]; -- Start keyboard process

STOP;

[] ← loaderDefs.loadBcd[NIL ! ANY => CONTINUE]; -- make loader fill in SD
loadSystems[! CoreSwapDefs.CAbort => CONTINUE];
```

END...