```
-- File: BootmesaDefs.Mesa
-- Last edited by Sandman; May 10, 1978  9:39 AM

DIRECTORY
  AltoDefs: FROM "altodefs",
  BcdDefs: FROM "bcddefs",
  ControlDefs: FROM "controldefs",
  FakeSegDefs: FROM "fakesegdefs",
  ImageDefs: FROM "imagedefs",
  LoaderBcdUtilDefs: FROM "loaderbcdutildefs",
  SegmentDefs: FROM "segmentdefs",
  StreamDefs: FROM "streamdefs",
  WartDefs: FROM "wartdefs";

DEFINITIONS FROM AltoDefs, FakeSegDefs, ControlDefs;

BootmesaDefs: DEFINITIONS SHARES ImageDefs =
  BEGIN

  BootIndex: TYPE = WartDefs.BootIndex;
  BootScriptHeader: TYPE = WartDefs.BootScriptHeader;

  FirstFrameWord: TYPE = MACHINE DEPENDENT RECORD [
    gfi: GFTIndex,
    unused: [0..3],
    alloced, shared, started: BOOLEAN,
    trapxfers, codelinks: BOOLEAN];

  -- From BootScript

  GetBootIndex: PROCEDURE RETURNS [BootIndex];
  GetBootTable: PROCEDURE RETURNS [CARDINAL];
  InitializeBootScript: PROCEDURE;
  FinishBootScript: PROCEDURE [startframe: GlobalFrameHandle]
    RETURNS [headerloc: Address];
  DeclareVMBounds: PROCEDURE [b: VMBounds];
  DeclareBootCommand: PROCEDURE [c: WartDefs.BootCommand];
  DeclareSwapOut: PROCEDURE [f: GlobalFrameHandle];
  DeclareOpenFiles: PROCEDURE;
  DeclareSegment: PROCEDURE [s: FakeSegmentHandle] RETURNS [BOOLEAN];
  DeclareCodeLink: PROCEDURE [f: GlobalFrameHandle] RETURNS [BOOLEAN];
  DeclareUnlock: PROCEDURE [s: FakeSegmentHandle];
  UnlockFrame: PROCEDURE [f: GlobalFrameHandle];
  WriteScriptSegments: PROCEDURE;

  -- From BootImage

  WriteSwappedIn: PROCEDURE [s: FakeSegmentHandle] RETURNS [BOOLEAN];
  WriteSwappedOut: PROCEDURE [s: FakeSegmentHandle] RETURNS [BOOLEAN];
  SegmentToMap: PROCEDURE [FakeSegmentHandle];
  EnterMapSegment: PROCEDURE [FakeSegmentHandle];
  DeclareSegments: PROCEDURE;

  -- From Bootmesa

  InitializeHeap: PROCEDURE;
  AllocGlobalFrame: PROCEDURE [
    framesize, nlinks: CARDINAL, framelinks: BOOLEAN] RETURNS [POINTER];
  DeclareLoadStateParameters: PROCEDURE [
    FakeSegmentHandle, FakeSegmentHandle, FakeSegmentHandle];
  AdjustBcd: PROCEDURE [FakeSegmentHandle];
  InitializeBootmesa: PROCEDURE [PageCount, STRING]
    RETURNS [BcdDefs.VersionStamp];
  InitializeImage: PROCEDURE;
  ListResidentCodeOffsets: PROCEDURE;
  LockFrame: PROCEDURE [GlobalFrameHandle];
  SetExtraFrames: PROCEDURE;
  SwapInFrame: PROCEDURE[GlobalFrameHandle];
  WriteImage: PROCEDURE;
  XFER: PROCEDURE [dest: GlobalFrameHandle];
  SetDefaultMemoryLimits: PROCEDURE [fp, lp: PageNumber];
  SetDefaultNProcesses: PROCEDURE [n: CARDINAL];
  SetDefaultGFTLength: PROCEDURE [l: CARDINAL];

  BootmesaError: PROCEDURE [msg: STRING];
  BootmesaModuleError: PROCEDURE [msg, config, module: STRING];
```

```
-- From BootLoader

Load: PROCEDURE [STRING]
  RETURNS [FakeSegmentHandle, FakeSegmentHandle, FakeSegmentHandle];
CantFindCodeFile: SIGNAL;
InvalidBcd: ERROR [bcdfile: SegmentDefs.FileHandle];
InvalidFile: ERROR [name: STRING];
OpenLoadmap: PROCEDURE [root: STRING];
CloseLoadmap: PROCEDURE;
InitializeGFT: PROCEDURE [POINTER, CARDINAL];


-- From BootUtilities

ModuleName: PROCEDURE [frame: GlobalFrameHandle, name: STRING];
Frame: PROCEDURE [STRING] RETURNS [GlobalFrameHandle];
InitUtilities: PROCEDURE [LoaderBcdUtilDefs.BcdBase];
SetConfig: PROCEDURE [name: STRING];
ResetConfig: PROCEDURE;

-- From Other Modules

-- From BootUser

OpenSource: PROCEDURE [root: STRING];
CloseSource: PROCEDURE;
ParseInput: PROCEDURE;
NubFrame, WartFrame, UserControl: PROCEDURE RETURNS [GlobalFrameHandle];
LookUpResidentModules: PROCEDURE;
EnumerateResidentModules: PROCEDURE [
  proc: PROCEDURE [GlobalFrameHandle] RETURNS [BOOLEAN]]
  RETURNS [GlobalFrameHandle];
TurnOffStartTrap: PROCEDURE;
NumberSwappedIn: PROCEDURE RETURNS [CARDINAL];
SwapInUserCode: PROCEDURE;
UnlockUserCode: PROCEDURE;

BootAbort: SIGNAL;

BuildNucleus: PROGRAM;

Bootmesa, BootImage, BootScript: PROGRAM [data: POINTER TO BootData];

BootData: TYPE = RECORD [
  imageFileRoot: STRING,
  imageFile: SegmentDefs.FileHandle,
  imageStream: StreamDefs.StreamHandle,
  image: POINTER TO ImageDefs.ImageHeader,
  vmFile: SegmentDefs.FileHandle,
  bsheader: POINTER TO BootScriptHeader,
  headerSeg: SegmentDefs.FileSegmentHandle,
  residentCodeSeg: FakeSegmentHandle,
  vmTableSeg: FakeSegmentHandle,
  AVbase, GFTbase, SDbase: POINTER,
  tableBase: POINTER];

END...
```