

**DosL**

COLLABORATORS

	TITLE : DosL		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY		July 22, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>DosL</b>	<b>1</b>
1.1	DosL . . . . .	1
1.2	TMP:Modula-2/DosL.def . . . . .	2

# Chapter 1

## DosL

### 1.1 DosL

#### Prozeduren

AbortPkt	AddBuffers	AddDosEntry
AddPart	AddSegment	AllocDosObject
AssignAdd	AssignLate	AssignLock
AssignPath	AttemptLockDosList	ChangeMode
CheckSignal	Cli	CliInitNewcli
CliInitRun	Close	CloseB
CompareDates	CreateDir	CreateNewProc
CreateProc	CurrentDir	DateStamp
DateToStr	Delay	DeleteFile
DeleteVar	DeviceProc	DoPkt
DupLock	DupLockFromFH	EndNotify
ErrorReport	ExAll	ExNext
Examine	ExamineFH	Execute
Exit	FGetC	FGets
FPutC	FPutS	FRead
FWrite	Fault	FilePart
FindArg	FindCliProc	FindDosEntry
FindSegment	FindVar	Flush
Format	FreeArgs	FreeDeviceProc
FreeDosEntry	FreeDosObject	GetArgStr
GetConsoleTask	GetCurrentDirName	GetDeviceProc
GetFileSysTask	GetProgramDir	GetProgramName
GetPrompt	GetVar	Info
Inhibit	Input	InternalLoadSeg
InternalUnLoadSeg	IoErr	IsFileSystem
IsInteractive	LoadSeg	Lock
LockDosList	LockRecord	LockRecords
MakeDosEntry	MakeHardLink	MakeLink
MakeSoftLink	MatchEnd	MatchFirst
MatchNext	MatchPattern	MatchPatternNoCase
MaxCli	NameFromFH	NameFromLock
NewLoadSeg	NextDosEntry	Open
OpenFromLock	Output	ParentDir
ParentOfFH	ParsePattern	ParsePatternNoCase
PathPart	PrintFault	PutStr
Read	ReadArgs	ReadItem

ReadLink	Relabel	RemAssignList
RemDosEntry	RemSegment	Rename
ReplyPkt	RunCommand	SameDevice
SameLock	Seek	SelectInput
SelectOutput	SendPkt	SetArgStr
SetComment	SetConsoleTask	SetCurrentDirName
SetFileDate	SetFileSize	SetFileSysTask
SetIoErr	SetMode	SetProgramDir
SetProgramName	SetPrompt	SetProtection
SetVBuf	SetVar	SplitName
StartNotify	StrToDate	StrToLong
System	SystemTagList	UnGetC
UnLoadSeg	UnLock	UnLockRecord
UnLockRecords	UnlockDosList	VFPprintf
VFWritef	VPrintf	WaitForChar
WaitPkt	Write	WriteChars

#### Konstanten

dosMinVersion	dosName
---------------	---------

#### Variablen

dosBase	dosVersion
---------	------------

## 1.2 TMP:Modula-2/DosL.def

```

DEFINITION MODULE DosL {"dos.library", 33};
(* 14-Apr-1992/cn *)

FROM SYSTEM IMPORT ADDRESS, BPTR, LONGSET;

(*
FROM SYSTEM IMPORT ADDRESS, BPTR, BYTE, LONGSET;

FROM ExecD IMPORT
  MsgPortPtr, Node, SignalSemaphore, Task;
FROM Timer IMPORT
  TimeRequestPtr;
FROM UtilityD IMPORT
  tagUser, TagItemPtr;
*)

FROM ExecD IMPORT
  MsgPortPtr ;

FROM UtilityD IMPORT
  TagItemPtr ;

IMPORT
  d: DosD, R;

VAR
  dosVersion: INTEGER;

```

---

```
dosBase:d. DosLibraryPtr ;

CONST
dosMinVersion=33;
dosName="dos.library";

PROCEDURE (*36*) AbortPkt (
    port{R.D1}:d. ProcessId ;
    pkt{R.D2}:d. DosPacketPtr ); CODE -264;

PROCEDURE (*36*) AddBuffers (
    name{R.D1}:d. StrPtr ;
    number{R.D2}:LONGINT):BOOLEAN; CODE -732;

PROCEDURE (*36*) AddDosEntry (
    dlist{R.D1}:d. DosListPtr ):BOOLEAN; CODE -678;

PROCEDURE (*36*) AddPart (
    dirname{R.D1}:d. StrPtr ;
    filename{R.D2}:d. StrPtr ;
    size{R.D3}:LONGINT):BOOLEAN; CODE -882;

PROCEDURE (*36*) AddSegment (
    name{R.D1}:d. StrPtr ;
    seg{R.D2}:d. SegmentPtr ;
    system{R.D3}:LONGINT):BOOLEAN; CODE -774;

PROCEDURE (*36*) AllocDosObject (
    type{R.D1}:LONGCARD;
    tags{R.D2}: TagItemPtr ):ADDRESS; CODE -228;

PROCEDURE (*36*) AssignAdd (
    name{R.D1}:d. StrPtr ;
    lock{R.D2}:d. FileLockPtr ):BOOLEAN; CODE -630;

PROCEDURE (*36*) AssignLate (
    name{R.D1}:d. StrPtr ;
    path{R.D2}:d. StrPtr ):BOOLEAN; CODE -618;

PROCEDURE (*36*) AssignLock (
    name{R.D1}:d. StrPtr ;
    lock{R.D2}:d. FileLockPtr ):BOOLEAN; CODE -612;

PROCEDURE (*36*) AssignPath (
    name{R.D1}:d. StrPtr ;
    path{R.D2}:d. StrPtr ):BOOLEAN; CODE -624;

PROCEDURE (*36*) AttemptLockDosList (
    flags{R.D1}:d. LockDosFlagSet ):d. DosListPtr ; CODE -666;

PROCEDURE (*37*) ChangeMode (
    type{R.D1}:LONGCARD;
    object{R.D2}:BPTR;
    newmode{R.D3}:LONGINT):BOOLEAN; CODE -450;

PROCEDURE (*36*) CheckSignal (mask{R.D1}:LONGSET):LONGSET; CODE -792;
```

---

```
PROCEDURE(*36*) Cli ():d. CommandLineInterfaceAPtr ; CODE -492;

PROCEDURE(*36*) CliInitNewcli (
    dp{R.A0}:d. DosPacketPtr ):d. CliInitFlagSet ; CODE -930;

PROCEDURE(*36*) CliInitRun (
    dp{R.A0}:d. DosPacketPtr ):d. CliInitFlagSet ; CODE -936;

PROCEDURE Close (file{R.D1}:d. FileHandlePtr ); CODE -36;

PROCEDURE(*36*) CloseB(file{R.D1}:d. FileHandlePtr ):BOOLEAN; CODE -36;
(*)
    CloseB ist eigentlich identisch mit Close. Aber ab WB2.0 hat Close
    ein Resultatwert, der Angibt, ob die Operations ausgeführt werden
    konnte oder nicht.
*)

PROCEDURE(*36*) CompareDates (
    date1{R.D1},date2{R.D2}:d. DatePtr ):LONGINT; CODE -738;

PROCEDURE CreateDir (name{R.D1}:d. StrPtr ):d. FileLockPtr ; CODE -120;

PROCEDURE(*36*) CreateNewProc (
    tags{R.D1}: TagItemPtr ):d. ProcessPtr ; CODE -498;

PROCEDURE CreateProc (
    name{R.D1}:d. StrPtr ;
    pri{R.D2}:LONGINT;
    segList{R.D3}:BPTR;
    stackSize{R.D4}:LONGINT):d. ProcessId ; CODE -138;

PROCEDURE CurrentDir (
    lock{R.D1}:d. FileLockPtr ):d. FileLockPtr ; CODE -126;

PROCEDURE DateStamp (v{R.D1}:d. DatePtr ); CODE -192;

PROCEDURE(*36*) DateToStr (
    datetime{R.D1}:d. DateTimePtr ):LONGINT; CODE -744;

PROCEDURE Delay (ticks{R.D1}:LONGINT); CODE -198;

PROCEDURE DeleteFile (name{R.D1}:d. StrPtr ):BOOLEAN; CODE -72;

PROCEDURE(*36*) DeleteVar (
    name{R.D1}:d. StrPtr ;
    flags{R.D2}:d. VarFlagSet ):BOOLEAN; CODE -912;

PROCEDURE DeviceProc (name{R.D1}:d. StrPtr ):d. ProcessId ; CODE -174;

PROCEDURE(*36*) DoPkt (
    port{R.D1}:d. ProcessId ;
    action{R.D2}:LONGINT;
    arg1{R.D3},arg2{R.D4},arg3{R.D5}:LONGINT;
    arg4{R.D6},arg5{R.D7}:LONGINT):LONGINT; CODE -240;

PROCEDURE DupLock (lock{R.D1}:d. FileLockPtr ):d. FileLockPtr ; CODE -96;
```

---

```
PROCEDURE (*36*) DupLockFromFH (
    fh{R.D1}:d. FileHandlePtr ):d. FileLockPtr ; CODE -372;

PROCEDURE (*36*) EndNotify (VAR notify{R.D1}:d. NotifyRequest ); CODE -894;

PROCEDURE (*36*) ErrorReport (
    code{R.D1}:LONGINT;
    type{R.D2}:LONGINT;
    arg1{R.D3}:LONGCARD;
    device{R.D4}:d. ProcessId ):BOOLEAN; CODE -480;

PROCEDURE (*36*) ExAll (
    lock{R.D1}:d. FileLockPtr ;
    buffer{R.D2}:ADDRESS;
    size{R.D3}:LONGINT;
    type{R.D4}:LONGINT;
    control{R.D5}:d. ExAllControlPtr ):BOOLEAN; CODE -432;

PROCEDURE Examine (
    lock{R.D1}:d. FileLockPtr ;
    infoBlock{R.D2}:d. FileInfoBlockPtr ):BOOLEAN; CODE -102;

PROCEDURE (*36*) ExamineFH (
    fh{R.D1}:d. FileHandlePtr ;
    fib{R.D2}:d. FileInfoBlockPtr ):BOOLEAN; CODE -390;

PROCEDURE Execute (
    commandString{R.D1}:d. StrPtr ;
    input{R.D2}:d. FileHandlePtr ;
    output{R.D3}:d. FileHandlePtr ):LONGINT; CODE -222;

PROCEDURE Exit (returnCode{R.D1}:LONGINT); CODE -144;

PROCEDURE ExNext (
    lock{R.D1}:d. FileLockPtr ;
    infoBlock{R.D2}:d. FileInfoBlockPtr ):BOOLEAN; CODE -108;

PROCEDURE (*36*) Fault (
    code{R.D1}:LONGINT;
    header{R.D2}:d. StrPtr ;
    buffer{R.D3}:d. StrPtr ;
    len{R.D4}:LONGINT):BOOLEAN; CODE -468;

PROCEDURE (*36*) FGetC (fh{R.D1}:d. FileHandlePtr ):LONGINT; CODE -306;

PROCEDURE (*36*) FGets (
    fh{R.D1}:d. FileHandlePtr ;
    buf{R.D2}:ADDRESS;
    buflen{R.D3}:LONGCARD):d. StrPtr ; CODE -336;

PROCEDURE (*36*) FilePart (path{R.D1}:d. StrPtr ):d. StrPtr ; CODE -870;

PROCEDURE (*36*) FindArg (
    keyword{R.D1}:d. StrPtr ;
    template{R.D2}:d. StrPtr ):LONGINT; CODE -804;

PROCEDURE (*36*) FindCliProc (num{R.D1}:LONGINT):d. ProcessPtr ; CODE -546;
```

---



```
PROCEDURE (*36*) FindDosEntry (
    dlist{R.D1}:d. DosListPtr ;
    name{R.D2}:d. StrPtr ;
    flags{R.D3}:d. LockDosFlagSet ):d. DosListPtr ; CODE -684;

PROCEDURE (*36*) FindSegment (
    name{R.D1}:d. StrPtr ;
    seg{R.D2}:d. ResidentSegmentAPtr ;
    system{R.D3}:BOOLEAN):d. ResidentSegmentAPtr ; CODE -780;

PROCEDURE (*36*) FindVar (
    name{R.D1}:d. StrPtr ;
    type{R.D2}:d. VarFlagSet ):d. LocalVarPtr ; CODE -918;

PROCEDURE (*36*) Flush (fh{R.D1}:d. FileHandlePtr ):BOOLEAN; CODE -360;

PROCEDURE (*36*) Format (
    filesystem{R.D1}:d. StrPtr ;
    volumename{R.D2}:d. StrPtr ;
    dostype{R.D3}:LONGINT):BOOLEAN; CODE -714;

PROCEDURE (*36*) FPutC (
    fh{R.D1}:d. FileHandlePtr ;
    ch{R.D2}:LONGINT):LONGINT; CODE -312;

PROCEDURE (*36*) FPutS (
    fh{R.D1}:d. FileHandlePtr ;
    str{R.D2}:d. StrPtr ):BOOLEAN; CODE -342;

PROCEDURE (*36*) FRead (
    fh{R.D1}:d. FileHandlePtr ;
    buf{R.D2}:ADDRESS;
    blocklen{R.D3}:LONGCARD;
    blocks{R.D4}:LONGCARD):LONGINT; CODE -324;

PROCEDURE (*36*) FreeArgs (args{R.D1}:d. RArgsPtr ); CODE -858;

PROCEDURE (*36*) FreeDeviceProc (devproc{R.D1}:d. DevProcPtr ); CODE -648;

PROCEDURE (*36*) FreeDosEntry (dlist{R.D1}:d. DosListPtr ); CODE -702;

PROCEDURE (*36*) FreeDosObject (
    type{R.D1}:LONGCARD;
    ptr{R.D2}:ADDRESS); CODE -234;

PROCEDURE (*36*) FWrite (
    fh{R.D1}:d. FileHandlePtr ;
    buf{R.D2}:ADDRESS;
    blocklen{R.D3}:LONGCARD;
    blocks{R.D4}:LONGCARD):LONGINT; CODE -330;

PROCEDURE (*36*) GetArgStr ():d. StrPtr ; CODE -534;

PROCEDURE (*36*) GetConsoleTask ():d. ProcessId ; CODE -510;

PROCEDURE (*36*) GetCurrentDirName (
```

---

```
        buf{R.D1}:d. StrPtr ;
        len{R.D2}:LONGINT):BOOLEAN; CODE -564;

PROCEDURE(*36*) GetDeviceProc (
    name{R.D1}:d. StrPtr ;
    devproc{R.D2}:d. DevProcPtr ):d. DevProcPtr ; CODE -642;

PROCEDURE(*36*) GetFileSysTask ():d. ProcessId ; CODE -522;

PROCEDURE(*36*) GetProgramDir ():d. FileLockPtr ; CODE -600;

PROCEDURE(*36*) GetProgramName (
    buf{R.D1}:d. StrPtr ;
    len{R.D2}:LONGINT):BOOLEAN; CODE -576;

PROCEDURE(*36*) GetPrompt (
    buf{R.D1}:d. StrPtr ;
    len{R.D2}:LONGINT):BOOLEAN; CODE -588;

PROCEDURE(*36*) GetVar (
    name{R.D1}:d. StrPtr ;
    buffer{R.D2}:d. StrPtr ;
    size{R.D3}:LONGINT;
    flags{R.D4}:d. VarFlagSet ):LONGINT; CODE -906;

PROCEDURE Info (
    lock{R.D1}:d. FileLockPtr ;
    parameterBlock{R.D2}:d. InfoDataPtr ):BOOLEAN; CODE -114;

PROCEDURE(*36*) Inhibit (
    name{R.D1}:d. StrPtr ;
    onoff{R.D2}:BOOLEAN):BOOLEAN; CODE -726;

PROCEDURE Input ():d. FileHandlePtr ; CODE -54;

PROCEDURE(*36*) InternalLoadSeg (
    fh{R.D0}:d. FileHandlePtr ;
    table{R.A0}:BPTR;
    funcarray{R.A1}:d. IArrayPtr ;
    VAR stack{R.A2}:LONGINT):BPTR; CODE -756;

PROCEDURE(*36*) InternalUnLoadSeg (
    seglist{R.D1}:BPTR;
    freefunc{R.A1}:d. FreeProc ):BOOLEAN; CODE -762;

PROCEDURE IoErr ():LONGINT; CODE -132;

PROCEDURE(*36*) IsFileSystem (name{R.D1}:d. StrPtr ):BOOLEAN; CODE -708;

PROCEDURE IsInteractive (file{R.D1}:d. FileHandlePtr ):BOOLEAN; CODE -216;

PROCEDURE LoadSeg (name{R.D1}:d. StrPtr ):BPTR; CODE -150;

PROCEDURE Lock (
    name{R.D1}:d. StrPtr ;
    accessMode{R.D2}:LONGINT):d. FileLockPtr ; CODE -84;
```

---

```

PROCEDURE (*36*) LockDosList (
    flags{R.D1}:d. LockDosFlagSet ):d. DosListPtr ; CODE -654;

PROCEDURE (*36*) LockRecord (
    fh{R.D1}:d. FileHandlePtr ;
    offset{R.D2},length{R.D3}:LONGCARD;
    mode{R.D4},timeout{R.D5}:LONGCARD):BOOLEAN; CODE -270;

PROCEDURE (*36*) LockRecords (
    recArr{R.D1}:d. RecordLockArrayPtr ;
    timeout{R.D2}:LONGCARD):BOOLEAN; CODE -276;

PROCEDURE (*36*) MakeDosEntry (
    name{R.D1}:d. StrPtr ;
    type{R.D2}:LONGINT):d. DosListPtr ; CODE -696;

PROCEDURE (*36*) MakeLink (
    name{R.D1}:d. StrPtr ;
    dest{R.D2}:LONGCARD;
    soft{R.D3}:BOOLEAN):LONGINT; CODE -444;

(*
Die Prozedur MakeLink kann sowohl für Soft- wie auch für Hardlinks
verwendet werden. Je nach dem muss dem Parameter dest die Adresse
eines Strings (soft) oder ein FileLockPtr übergeben werden. Dies
bedingt manchmal einen Typencast.

Für diejenigen Fälle, wo man nur Soft- oder nur Hardlinks erstellen
möchte, bieten sich die nächsten beiden Deklarationen an. Der Typ
von dest wurde den entsprechenden Erfordernissen angepasst. Man muss
aber aufpassen, den dritten Parameter (soft) richtig zu setzen, also
immer auf TRUE bei der Verwendung von MakeSoftLink und immer auf
FALSE bei MakeHardLink.
*)

PROCEDURE (*36*) MakeHardLink(
    name{R.D1}:d. StrPtr ;
    dest{R.D2}:d. FileLockPtr ;
    soft{R.D3}:BOOLEAN (*FALSE*)):LONGINT; CODE -444;

PROCEDURE (*36*) MakeSoftLink(
    name{R.D1}:d. StrPtr ;
    dest{R.D2}:d. StrPtr ;
    soft{R.D3}:BOOLEAN (*TRUE*)):LONGINT; CODE -444;

PROCEDURE (*36*) MatchEnd (VAR anchor{R.D1}:d. AnchorPath ); CODE -834;

PROCEDURE (*36*) MatchFirst (
    pat{R.D1}:d. StrPtr ;
    VAR anchor{R.D2}:d. AnchorPath ):LONGINT; CODE -822;

PROCEDURE (*36*) MatchNext (
    VAR anchor{R.D1}:d. AnchorPath ):LONGINT; CODE -828;

PROCEDURE (*36*) MatchPattern (
    pat{R.D1}:d. StrPtr ;
    str{R.D2}:d. StrPtr ):BOOLEAN; CODE -846;

```

---

```
PROCEDURE (*37*) MatchPatternNoCase (
    pat{R.D1}:d. StrPtr ;
    str{R.D2}:d. StrPtr ):BOOLEAN; CODE -972;

PROCEDURE (*36*) MaxCli ():LONGINT; CODE -552;

PROCEDURE (*36*) NameFromFH (
    fh{R.D1}:d. FileHandlePtr ;
    buffer{R.D2}:d. StrPtr ;
    len{R.D3}:LONGINT):BOOLEAN; CODE -408;

PROCEDURE (*36*) NameFromLock (
    lock{R.D1}:d. FileLockPtr ;
    buffer{R.D2}:d. StrPtr ;
    len{R.D3}:LONGINT):BOOLEAN; CODE -402;

PROCEDURE (*36*) NewLoadSeg (
    file{R.D1}:d. StrPtr ;
    tags{R.D2}: TagItemPtr ):BPTR; CODE -768;

PROCEDURE (*36*) NextDosEntry (
    dlist{R.D1}:d. DosListPtr ;
    flags{R.D2}:d. LockDosFlagSet ):d. DosListPtr ; CODE -690;

PROCEDURE Open (
    name{R.D1}:d. StrPtr ;
    accessMode{R.D2}:LONGINT):d. FileHandlePtr ; CODE -30;

PROCEDURE (*36*) OpenFromLock (
    lock{R.D1}:d. FileLockPtr ):d. FileHandlePtr ; CODE -378;

PROCEDURE Output ():d. FileHandlePtr ; CODE -60;

PROCEDURE ParentDir (lock{R.D1}:d. FileLockPtr ):d. FileLockPtr ; CODE -210;

PROCEDURE (*36*) ParentOfFH (
    fh{R.D1}:d. FileHandlePtr ):d. FileLockPtr ; CODE -384;

PROCEDURE (*36*) ParsePattern (
    pat{R.D1}:d. StrPtr ;
    buf{R.D2}:d. StrPtr ;
    buflen{R.D3}:LONGINT):LONGINT; CODE -840;

PROCEDURE (*37*) ParsePatternNoCase (
    pat{R.D1}:d. StrPtr ;
    buf{R.D2}:d. StrPtr ;
    buflen{R.D3}:LONGINT):LONGINT; CODE -966;

PROCEDURE (*36*) PathPart (path{R.D1}:d. StrPtr ):d. StrPtr ; CODE -876;

PROCEDURE (*36*) PrintFault (
    code{R.D1}:LONGINT;
    header{R.D2}:d. StrPtr ):BOOLEAN; CODE -474;

PROCEDURE (*36*) PutStr (str{R.D1}:d. StrPtr ):BOOLEAN; CODE -948;

PROCEDURE Read (
```

---

```
    file{R.D1}:d. FileHandlePtr ;
    buffer{R.D2}:ADDRESS;
    length{R.D3}:LONGINT):LONGINT; CODE -42;

PROCEDURE (*36*)  ReadArgs (
    template{R.D1}:d. StrPtr ;
    array{R.D2}:ADDRESS;
    args{R.D3}:d. RArgsPtr ):d. RArgsPtr ; CODE -798;

PROCEDURE (*36*)  ReadItem (
    name{R.D1}:d. StrPtr ;
    maxchars{R.D2}:LONGINT;
    cSource{R.D3}:d. CSourcePtr ):LONGINT; CODE -810;

PROCEDURE (*36*)  ReadLink (
    port{R.D1}:d. ProcessId ;
    lock{R.D2}:d. FileLockPtr ;
    path{R.D3}:d. StrPtr ;
    buffer{R.D4}:d. StrPtr ;
    buffSize{R.D5}:LONGCARD):BOOLEAN; CODE -438;

PROCEDURE (*36*)  Relabel (
    drive{R.D1}:d. StrPtr ;
    newname{R.D2}:d. StrPtr ):BOOLEAN; CODE -720;

PROCEDURE (*36*)  RemAssignList (
    name{R.D1}:d. StrPtr ;
    lock{R.D2}:d. FileLockPtr ):BOOLEAN; CODE -636;

PROCEDURE (*36*)  RemDosEntry (
    dlist{R.D1}:d. DosListPtr ):BOOLEAN; CODE -672;

PROCEDURE (*36*)  RemSegment (seg{R.D1}:d. ResidentSegmentAPtr ):BOOLEAN; CODE ←
    -786;

PROCEDURE  Rename (
    oldName{R.D1},newName{R.D2}:d. StrPtr ):BOOLEAN; CODE -78;

PROCEDURE (*36*)  ReplyPkt (
    dp{R.D1}:d. DosPacketPtr ;
    res1{R.D2},res2{R.D3}:LONGINT); CODE -258;

PROCEDURE (*36*)  RunCommand (
    seg{R.D1}:BPTR;
    stacksize{R.D2}:LONGINT;
    paramptr{R.D3}:d. StrPtr ;
    paramlen{R.D4}:LONGINT):LONGINT; CODE -504;

PROCEDURE (*37*)  SameDevice (
    lock1{R.D1}:d. FileLockPtr ;
    lock2{R.D2}:d. FileLockPtr ):BOOLEAN; CODE -984;

PROCEDURE (*36*)  SameLock (
    lock1{R.D1}:d. FileLockPtr ;
    lock2{R.D2}:d. FileLockPtr ):LONGINT; CODE -420;

PROCEDURE  Seek (
```

---

```
    file{R.D1}:d. FileHandlePtr ;
    position{R.D2}:LONGINT;
    mode{R.D3}:LONGINT):LONGINT; CODE -66;

PROCEDURE(*36*) SelectInput (
    fh{R.D1}:d. FileHandlePtr ):d. FileHandlePtr ; CODE -294;

PROCEDURE(*36*) SelectOutput (
    fh{R.D1}:d. FileHandlePtr ):d. FileHandlePtr ; CODE -300;

PROCEDURE(*36*) SendPkt (
    dp{R.D1}:d. ProcessId ;
    port{R.D2}: MsgPortPtr ;
    replyport{R.D3}: MsgPortPtr ); CODE -246;

PROCEDURE(*36*) SetArgStr (string{R.D1}:d. StrPtr ):d. StrPtr ; CODE -540;

PROCEDURE SetComment (
    name{R.D1},comment{R.D2}:d. StrPtr ):BOOLEAN; CODE -180;

PROCEDURE(*36*) SetConsoleTask (
    task{R.D1}:d. ProcessId ):d. ProcessId ; CODE -516;

PROCEDURE(*36*) SetCurrentDirName (
    name{R.D1}:d. StrPtr ):BOOLEAN; CODE -558;

PROCEDURE(*36*) SetFileDate (
    name{R.D1}:d. StrPtr ;
    date{R.D2}:d. DatePtr ):BOOLEAN; CODE -396;

PROCEDURE(*36*) SetFileSize (
    fh{R.D1}:d. FileHandlePtr ;
    pos{R.D2},mode{R.D3}:LONGINT):LONGINT; CODE -456;

PROCEDURE(*36*) SetFileSysTask (
    task{R.D1}:d. ProcessId ):d. ProcessId ; CODE -528;

PROCEDURE(*36*) SetIoErr (result{R.D1}:LONGINT):LONGINT; CODE -462;

PROCEDURE(*36*) SetMode (
    fh{R.D1}:d. FileHandlePtr ;
    mode{R.D2}:LONGINT):BOOLEAN; CODE -426;

PROCEDURE(*36*) SetProgramDir (
    lock{R.D1}:d. FileLockPtr ):d. FileLockPtr ; CODE -594;

PROCEDURE(*36*) SetProgramName (name{R.D1}:d. StrPtr ):BOOLEAN; CODE -570;

PROCEDURE(*36*) SetPrompt (name{R.D1}:d. StrPtr ):BOOLEAN; CODE -582;

PROCEDURE SetProtection (
    name{R.D1}:d. StrPtr ;
    mask{R.D2}:d. ProtectionFlagSet ):BOOLEAN; CODE -186;

PROCEDURE(*36*) SetVar (
    name{R.D1}:d. StrPtr ;
    buffer{R.D2}:d. StrPtr ;
```

---

```

        size{R.D3}:LONGINT;
        flags{R.D4}:d. VarFlagSet ):BOOLEAN; CODE -900;

PROCEDURE (*36*) SetVBuf (
    fh{R.D1}:d. FileHandlePtr ;
    buff{R.D2}:ADDRESS;
    type{R.D3}:LONGINT;
    size{R.D4}:LONGINT):BOOLEAN; CODE -366;

PROCEDURE (*36*) SplitName (
    name{R.D1}:d. StrPtr ;
    seperator{R.D2}:CHAR;
    buf{R.D3}:ADDRESS;
    oldpos{R.D4}:LONGINT;
    size{R.D5}:LONGINT):LONGINT; CODE -414;

PROCEDURE (*36*) StartNotify (
    VAR notify{R.D1}:d. NotifyRequest ):BOOLEAN; CODE -888;

PROCEDURE (*36*) StrToDate (
    datetime{R.D1}:d. DateTimePtr ):LONGINT; CODE -750;

PROCEDURE (*36*) StrToLong (
    string{R.D1}:d. StrPtr ;
    VAR value{R.D2}:LONGINT):LONGINT; CODE -816;

PROCEDURE (*36*) System(
    command{R.D1}:d. StrPtr ;
    tags{R.D2}: TagItemPtr ):LONGINT; CODE -606;

(*
  Die Prozedur System wird in der Commodore Dokumentation auch
  SystemTagList genannt. Die nächste Deklaration ist bis auf den
  Namen identisch mit System.
*)
PROCEDURE (*36*) SystemTagList (
    command{R.D1}:d. StrPtr ;
    tags{R.D2}: TagItemPtr ):LONGINT; CODE -606;

PROCEDURE (*36*) UnGetC (
    fh{R.D1}:d. FileHandlePtr ;
    character{R.D2}:LONGINT):LONGINT; CODE -318;

PROCEDURE  UnLoadSeg (segment{R.D1}:d. SegmentPtr ); CODE -156;

PROCEDURE  UnLock (lock{R.D1}:d. FileLockPtr ); CODE -90;

PROCEDURE (*36*) UnlockDosList (flags{R.D1}:d. LockDosFlagSet ); CODE -660;

PROCEDURE (*36*) UnLockRecord (
    fh{R.D1}:d. FileHandlePtr ;
    offset{R.D2},length{R.D3}:LONGCARD):BOOLEAN; CODE -282;

PROCEDURE (*36*) UnLockRecords (
    recArr{R.D1}:d. RecordLockArrayPtr ):BOOLEAN; CODE -288;

PROCEDURE (*36*) VFPrintf (
    fh{R.D1}:d. FileHandlePtr ;

```

---

```
        format{R.D2}:d. StrPtr ;
        argarray{R.D3}:ADDRESS):LONGINT; CODE -354;

PROCEDURE(*36*) VFWritef (
    fh{R.D1}:d. FileHandlePtr ;
    format{R.D2}:d. StrPtr ;
    argarray{R.D3}:ADDRESS):LONGINT; CODE -348;

PROCEDURE(*36*) VPrintf (
    format{R.D1}:d. StrPtr ;
    argarray{R.D2}:ADDRESS):LONGINT; CODE -954;

PROCEDURE WaitForChar (
    file{R.D1}:d. FileHandlePtr ;
    timeout{R.D2}:LONGINT):BOOLEAN; CODE -204;

PROCEDURE(*36*) WaitPkt ():d. DosPacketPtr ; CODE -252;

PROCEDURE Write (
    file{R.D1}:d. FileHandlePtr ;
    buffer{R.D2}:ADDRESS;
    length{R.D3}:LONGINT):LONGINT; CODE -48;

PROCEDURE(*36*) WriteChars (
    buf{R.D1}:ADDRESS;
    buflen{R.D2}:LONGINT):LONGINT; CODE -942;

END DosL.lib33
```

---