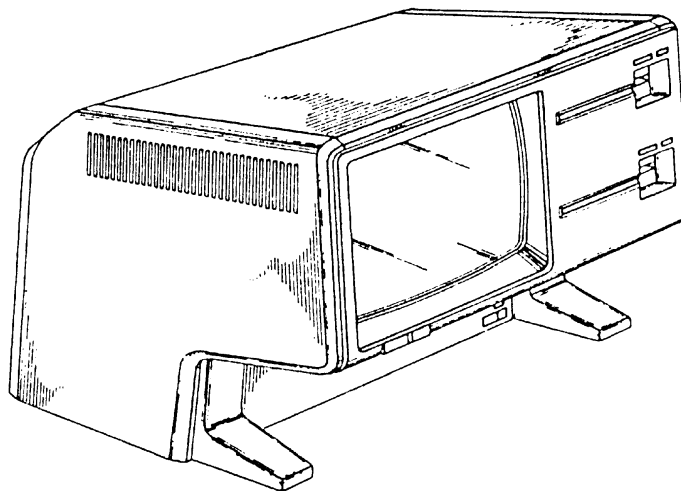


Apple Lisa

Operating System Library

Pascal Interface



This is a listing of the Pascal interface to the Lisa's operating system library. This OS provided a multi-tasking virtual-memory process protected architecture. The file system was hierarchical and supported file recovery via disk block tags. The file system also supported disk file labels which were small files that could be assigned to each file and which were a precursor to the Macintosh file resource fork concept.

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA OS INTERFACE.TEXT"
=====
```

```
000001 .....
000002 .....
000003 .....
000004 .....
000005                APPLE LISA OPERATING SYSTEM PASCAL INTERFACE
000006 .....
000007 .....
000008 .....
000009 .....
000010 .....
000011 USES {$U-} {$U syscall} SYSCALL;
000012 .....
000013 .....
000014                { system call definitions unit }
000015 INTRINSIC;
000016 .....
000017                { Copyright 1983, 1984, Apple Computer Inc. }
000018 .....
000019 INTERFACE
000020 .....
000021 CONST
000022     max_ename = 32;           { maximum length of a file system object name }
000023     max_pathname = 255;      { maximum length of a file system pathname }
000024     max_label_size = 128;    { maximum size of a file label, in bytes }
000025     len_exname = 16;         { length of exception name }
000026     size_exdata = 11;        { 48 bytes, exception data block should have the same
000027                               size as r_eventblk, received event block }
000028 .....
000029     size_etext = 9;           { event text size - 40 bytes }
000030     size_waitlist = 10;      { size of wait list - should be same as reqptr_list }
000031 .....
000032                               { exception kind definitions for 'SYS_TERMINATE'
000033                               exception }
000034     call_term = 0;           { process called terminate_process }
000035     ended      = 1;           { process executed 'end' statement }
000036     self_killed = 2;         { process called kill_process on self }
000037     killed     = 3;           { process was killed by another process }
000038     fthr_term  = 4;           { process's father is terminating }
000039     bad_syscall = 5;          { process made invalid sys call - subcode bad }
000040     bad_errnum = 6;           { process passed bad address for errnum parm }
000041     swap_error = 7;           { process aborted due to code swap-in error }
000042     stk_overflow = 8;         { process exceeded max size (+T nnn) of stack }
000043     data_overflow = 9;        { process tried to exceed max data space size }
000044     parity_err = 10;          { process got a parity error while executing }
000045 .....
000046     def_div_zero = 11;        { default handler for div zero exception was called }
000047     def_value_oob = 12;       { " for value oob exception }
000048     def_ovfw     = 13;       { " for overflow exception }
000049     def_nmi_key  = 14;       { " for NMI key exception }
000050     def_range    = 15;       { " for 'SYS_VALUE_OOB' excep due to value range err }
000051     def_str_index = 16;      { " for 'SYS_VALUE_OOB' excep due to STRING index err }
000052 .....
000053     bus_error = 21;           { bus error occurred }
000054     addr_error = 22;          { address error occurred }
000055     illg_inst = 23;           { illegal instruction trap occurred }
000056     priv_violation = 24;      { privilege violation trap occurred }
000057     line_1010 = 26;           { line 1010 emulator occurred }
000058     line_1111 = 27;           { line 1111 emulator occurred }
000059 .....
000060     unexpected_ex = 29;       { an unexpected exception occurred }
000061 .....
```

Apple Lisa Computer Technical Information

```
000062    div_zero    = 31;        { exception kind definitions for hardware exception }
000063    value_oob   = 32;
000064    ovfw        = 33;
000065    nmi_key     = 34;
000066    value_range = 35;        { excep kind for value range and STRING index error }
000067    str_index   = 36;        { Note that these two cause 'SYS_VALUE_OOB' excep }
000068
000069    {DEVICE_CONTROL FUNCTIONS}
000070
000071    dvParity = 1;            {RS-232}
000072    dvOutDTR = 2;          {RS-232}
000073    dvOutXON = 3;          {RS-232}
000074    dvOutDelay = 4;        {RS-232}
000075    dvBaud = 5;            {RS-232}
000076    dvInWait = 6;          {RS-232, CONSOLE}
000077    dvInDTR = 7;           {RS-232}
000078    dvInXON = 8;           {RS-232}
000079    dvTypeahd = 9;         {RS-232}
000080    dvDiscon = 10;         {RS-232}
000081    dvOutNoHS = 11;        {RS-232}
000082    dvErrStat = 15;        {PROFILE}
000083    dvGetEvent = 16;       {CONSOLE}
000084    dvAutoLF = 17;         {RS-232, CONSOLE, PARALLEL PRINTER} {not yet}
000085    dvDiskStat = 20;       {TWIGGY, PROFILE}
000086    dvDiskSpare = 21;      {TWIGGY, PROFILE}
000087
000088    {Generic 'slot' position numbers: }
000089    cd_slot1 = 1;
000090    cd_slot2 = 2;
000091    cd_slot3 = 3;
000092    cd_scc = 10;
000093    cd_paraport = 11;
000094    cd_intdisk = 12;
000095    cd_sony = 13;
000096    cd_twigggy = 14;
000097    cd_console = 15;
000098
000099    {cpu_board values (see minfo), showing which cpu board type is present: }
000100    cpub_lisa = 0;
000101    cpub_pepsi = 1;
000102
000103    {io_board values (see minfo), showing which io board type is present: }
000104    iob_lisa = 0;
000105    iob_pepsi = 1;
000106    iob_sony = 2;
000107
000108
000109    TYPE
000110    pathname    = STRING [max_pathname];
000111    e_name      = STRING [max_ename];
000112    namestring  = STRING [20];
000113
000114    procinfoRec = record
000115        proppathname : pathname;
000116        global_id    : LONGINT;
000117        father_id    : LONGINT;
000118        priority     : 1..255;
000119        state        : (pactive, psuspended, pwaiting);
000120        data_in      : BOOLEAN
000121    end;
000122
000123    Tdstype = (ds_shared, ds_private); { types of data segments }
000124
000125    dsinfoRec = record
000126        mem_size : LONGINT;
000127        disc_size : LONGINT;
```

Apple Lisa Computer Technical Information

```
000128         numb_open : INTEGER;
000129         ldsn       : INTEGER;
000130         boundF     : BOOLEAN;
000131         presentF   : BOOLEAN;
000132         creatorF   : BOOLEAN;
000133         raccess    : BOOLEAN;
000134         segptr     : LONGINT;
000135         volname    : e_name;
000136     end;
000137
000138
000139     t_ex_name = STRING [len_exname];           { exception name }
000140     longadr = ^LONGINT;
000141     t_ex_state = (enabled, queued, ignored);  { exception state }
000142     p_ex_data = ^t_ex_data;
000143     t_ex_data = array [0..size_exdata] of LONGINT; { exception data blk }
000144
000145     t_ex_sts = record                          { exception status }
000146         ex_occurred_f : BOOLEAN;              { exception occurred flag}
000147         ex_state : t_ex_state;                { exception state }
000148         num_excep : INTEGER;                  { number of exceptions q'ed}
000149         hdl_adr : longadr;                    { handler address }
000150     end;
000151
000152     p_env_blk = ^env_blk;
000153     env_blk = record                          { environment block to pass to handler }
000154         pc : LONGINT;                          { program counter }
000155         sr : INTEGER;                          { status register }
000156         d0 : LONGINT;                          { data registers 0 - 7 }
000157         d1 : LONGINT;
000158         d2 : LONGINT;
000159         d3 : LONGINT;
000160         d4 : LONGINT;
000161         d5 : LONGINT;
000162         d6 : LONGINT;
000163         d7 : LONGINT;
000164         a0 : LONGINT;                          { address registers 0 - 7 }
000165         a1 : LONGINT;
000166         a2 : LONGINT;
000167         a3 : LONGINT;
000168         a4 : LONGINT;
000169         a5 : LONGINT;
000170         a6 : LONGINT;
000171         a7 : LONGINT;
000172     end;
000173
000174     p_term_ex_data = ^term_ex_data;
000175
000176     term_ex_data = record                    { terminate exception data block }
000177         case excep_kind : LONGINT of
000178             call_term,
000179             ended,
000180             self_killed,
000181             killed,
000182             fthr_term,
000183             bad_syscall,
000184             bad_errnum,
000185             swap_error,
000186             stk_overflow,
000187             data_overflow,
000188             parity_err : ();                  { due to process termination }
000189
000190             illg_inst,
000191             priv_violation,                  { due to illegal instruction,
000192             line_1010,                       privilege violation }
000193
```

Apple Lisa Computer Technical Information

```
000194         line_1111,           { due to line 1010, 1111 emulator }
000195         def_div_zero,
000196         def_value_oob,
000197         def_ovfw,
000198         def_nmi_key           { terminate due to default handler for
000199                               hardware exception }
000200         : (sr : INTEGER;
000201           pc : LONGINT);      { at the time of occurrence }
000202         def_range,
000203         def_str_index        { terminate due to default handler for
000204                               'SYS_VALUE_OOB' excep for value
000205                               range or STRING index error }
000206         : (value_check : INTEGER;
000207           upper_bound : INTEGER;
000208           lower_bound : INTEGER;
000209           return_pc  : LONGINT;
000210           caller_a6  : LONGINT);
000211         bus_error,
000212         addr_error          { due to bus error or address error }
000213         : (fun_field : packed record      { one INTEGER }
000214           filler : 0..$7ff;             { 11 bits }
000215           r_w_flag : BOOLEAN;
000216           i_n_flag : BOOLEAN;
000217           fun_code : 0..7;             { 3 bits }
000218           end;
000219           access_adr : LONGINT;
000220           inst_register : INTEGER;
000221           sr_error : INTEGER;
000222           pc_error : LONGINT);
000223         end;
000224
000225         p_hard_ex_data = ^hard_ex_data;
000226         hard_ex_data = record           { hardware exception data block }
000227         case excep_kind : LONGINT of
000228         div_zero, value_oob, ovfw
000229         : (sr : INTEGER;
000230           pc : LONGINT);
000231         value_range, str_index
000232         : (value_check : INTEGER;
000233           upper_bound : INTEGER;
000234           lower_bound : INTEGER;
000235           return_pc  : LONGINT;
000236           caller_a6  : LONGINT);
000237         end;
000238
000239
000240         accesses = (dread, dwrite, append, private, global_refnum);
000241         mset = set of accesses;
000242         iomode = (absolute, relative, sequential);
000243
000244         UID = record {unique id}
000245         a,b: LONGINT
000246         end;
000247
000248         timestmp_interval = record      { time interval }
000249         sec : LONGINT;                 { number of seconds }
000250         msec : 0..999;                 { number of milliseconds within a second }
000251         end;
000252
000253         info_type = (device_t, volume_t, object_t);
000254         devtype = (diskdev, pascalbd, seqdev, bitbkt, non_io);
000255         filetype = (undefined, MDDFFile, rootcat, freelist, badblocks,
000256                   sysdata, spool, exec, usercat, pipe, bootfile,
000257                   swapdata, swapcode, ramap, userfile, killedobject);
000258
000259         entrytype= (emptyentry, catentry, linkentry, fileentry, pipeentry, ecentry,
```

Apple Lisa Computer Technical Information

```
000260         killedentry);
000261
000262     { Per-file version control information record }
000263     Build_Control = record
000264         release_number : INTEGER; { public release number }
000265         build_number   : INTEGER; { internal build membership }
000266         compatibility_level : INTEGER; { local compatibility level }
000267         revision_level  : INTEGER; { iteration of file }
000268     end;
000269
000270     fs_info = record
000271         name : e_name;
000272         dir_path : pathname;
000273         machine_id : LONGINT;
000274         fs_overhead : INTEGER;
000275         result_scavenge : INTEGER;
000276         passwd_present : BOOLEAN;
000277         case otype : info_type of
000278             device_t, volume_t: (
000279                 iochannel : INTEGER;
000280                 slot_no : INTEGER;
000281                 devicenumb: INTEGER;
000282                 devt : devtype;
000283                 ejectable, removable: BOOLEAN;
000284                 fs_size : LONGINT;
000285                 vol_size : LONGINT;
000286                 blockstructured, mounted : BOOLEAN;
000287                 opencount : LONGINT;
000288                 private_dev, remote, locked_dev : BOOLEAN;
000289                 mount_pending, unmount_pending : BOOLEAN;
000290                 volname, password : e_name;
000291                 fsversion, volnum : INTEGER;
000292                 volid : UID;
000293                 backup_volid : UID;
000294                 vol_sequence : INTEGER;
000295                 blocksize, datasize, clustersize, filecount : INTEGER;
000296                 label_size : INTEGER;
000297                 freecount : LONGINT;
000298                 DTVC, DTCC, DTVB, DTVS : LONGINT;
000299                 master_copy_id, copy_thread : LONGINT;
000300                 overmount_stamp : UID;
000301                 boot_code : INTEGER;
000302                 boot_environ : INTEGER;
000303                 privileged, write_protected : BOOLEAN;
000304                 master, copy, copy_flag, scavenge_flag : BOOLEAN;
000305                 vol_left_mounted : BOOLEAN );
000306
000307             object_t : (
000308                 size : LONGINT;
000309                 psize : LONGINT; { physical file size in bytes }
000310                 lpsize : INTEGER; { logical page size in bytes for this file }
000311                 ftype : filetype;
000312                 etype : entrytype;
000313                 DTC, DTA, DTM, DTB, DTS : LONGINT;
000314                 refnum : INTEGER;
000315                 fmark : LONGINT;
000316                 acmode : mset;
000317                 nreaders, nwriters, nusers : INTEGER;
000318                 fuid : UID;
000319                 user_type : INTEGER;
000320                 user_subtype : INTEGER;
000321                 system_type : INTEGER;
000322                 eof, safety_on, kswitch : BOOLEAN;
000323                 private, locked, protected, master_file : BOOLEAN;
000324                 file_scavenged, file_closed_by_OS, file_left_open : BOOLEAN;
000325                 file_portion : INTEGER;
```

Apple Lisa Computer Technical Information

```
000326         build_info : Build_Control )
000327     end;
000328
000329     Q_Info = record
000330         name : e_name;
000331         etype : entrytype;
000332         DTC : LONGINT;
000333         DTM : LONGINT;
000334         size : LONGINT;
000335         psize : LONGINT;
000336         fs_overhead : INTEGER;
000337         master : BOOLEAN;
000338         protected : BOOLEAN;
000339         safety : BOOLEAN;
000340         left_open : BOOLEAN;
000341         scavenged : BOOLEAN;
000342         closed_by_OS : BOOLEAN;
000343         nreaders : INTEGER;
000344         nwriters : INTEGER;
000345         level : INTEGER;
000346     end;
000347
000348     dctype = record
000349         dcversion : INTEGER;
000350         dccode : INTEGER;
000351         dcdata : array [0..9] of LONGINT; { user/driver defined data }
000352     end;
000353
000354     t_waitlist = record { wait list }
000355         length : INTEGER;
000356         refnum : array [0..size_waitlist] of INTEGER;
000357     end;
000358
000359     t_eheader = record { event header }
000360         send_pid : LONGINT; { sender's process id }
000361         event_type : LONGINT; { type of event }
000362     end;
000363
000364     t_event_text = array [0..size_etext] of LONGINT;
000365     p_r_eventblk = ^r_eventblk;
000366     r_eventblk = record
000367         event_header : t_eheader;
000368         event_text : t_event_text;
000369     end;
000370
000371     p_s_eventblk = ^s_eventblk;
000372     s_eventblk = t_event_text;
000373
000374     time_rec = record
000375         year : INTEGER;
000376         day : 1..366; { julian date }
000377         hour : -23..23;
000378         minute : -59..59;
000379         second : 0..59;
000380         msec : 0..999;
000381     end;
000382
000383     chn_kind = (wait_ec, call_ec);
000384     t_chn_sts = record { channel status }
000385         chn_type : chn_kind; { channel type }
000386         num_events : INTEGER; { number of events queued }
000387         open_rcv : INTEGER; { number of opens for receiving }
000388         open_send : INTEGER; { number of opens for sending }
000389         ec_name : pathname; { event channel name }
000390     end;
000391
```

Apple Lisa Computer Technical Information

```
000392
000393
000394 {configuration stuff: }
000395
000396     slot_array = array [1..3] of INTEGER;
000397
000398     minfo = record
000399         cpu_board, io_board, memsize: LONGINT;
000400     end; {of minfo}
000401
000402
000403 { Lisa Office System parameter memory type }
000404
000405     pmByte = -128..127;
000406     pMemRec = array[0..63] of pmByte;
000407
000408     cd_position = record
000409         slot: pmbyte;           {slot number}
000410         chan: pmbyte;          {channel number}
000411         dev: pmbyte;           {device number}
000412     end;
000413
000414     ConfigDev = record
000415         pos: cd_position;       {device position}
000416         nExtWords: pmbyte;      {number of valid extension words}
000417         Extwords: array[1..3] of INTEGER; {extension words}
000418         DriverID: LONGINT;      {ID of driver controlling device}
000419         DevName: e_name;        {device name}
000420     end;
000421
000422     cd_infobuf = record {defines internal driver image}
000423         cd_driverid: LONGINT;
000424         cd_perm: BOOLEAN;
000425         cd_drvrname: e_name;
000426         case cd_devt: devtype of
000427             diskdev:
000428                 (cd_start_block: LONGINT;
000429                  cd_fs_start: LONGINT;
000430                  cd_ejectable: BOOLEAN;
000431                  cd_removable: BOOLEAN;
000432                  cd_preload: BOOLEAN;)
000433         end; {of cd_infobuf}
000434
000435
000436
000437
000438 { File System calls }
000439
000440     PROCEDURE MAKE_FILE (VAR ecode:INTEGER; VAR path:pathname; label_size:INTEGER);
000441
000442     PROCEDURE MAKE_PIPE (VAR ecode:INTEGER; VAR path:pathname; label_size:INTEGER);
000443
000444     PROCEDURE MAKE_CATALOG (VAR ecode:INTEGER; VAR path:pathname; label_size:INTEGER);
000445
000446     PROCEDURE MAKE_LINK (VAR ecode:INTEGER; VAR path, ref:pathname; label_size:INTEGER);
000447
000448     PROCEDURE KILL_OBJECT (VAR ecode:INTEGER; VAR path:pathname);
000449
000450     PROCEDURE UNKILL_FILE (VAR ecode:INTEGER; refnum:INTEGER; VAR new_name:e_name);
000451
000452     PROCEDURE OPEN (VAR ecode:INTEGER; VAR path:pathname; VAR refnum:INTEGER; manip:mset);
000453
000454     PROCEDURE CLOSE_OBJECT (VAR ecode:INTEGER; refnum:INTEGER);
000455
000456     PROCEDURE READ_DATA (VAR ecode : INTEGER;
000457                         refnum : INTEGER;
```


Apple Lisa Computer Technical Information

```
000458         data_addr : LONGINT;
000459         count : LONGINT;
000460     VAR actual : LONGINT;
000461         mode : iomode;
000462         offset : LONGINT);
000463
000464     PROCEDURE WRITE_DATA (VAR ecode : INTEGER;
000465         refnum : INTEGER;
000466         data_addr : LONGINT;
000467         count : LONGINT;
000468         VAR actual : LONGINT;
000469         mode : iomode;
000470         offset : LONGINT);
000471
000472     PROCEDURE FLUSH (VAR ecode:INTEGER; refnum:INTEGER);
000473
000474     PROCEDURE LOOKUP (VAR ecode : INTEGER;
000475         VAR path : pathname;
000476         VAR attributes : fs_info);
000477
000478     PROCEDURE INFO (VAR ecode:INTEGER; refnum:INTEGER; VAR refinfo:fs_info);
000479
000480     PROCEDURE QUICK_LOOKUP ( VAR ecode    : INTEGER;
000481         VAR path    : pathname;
000482         VAR InfoRec : Q_Info );
000483
000484     PROCEDURE ALLOCATE (VAR ecode : INTEGER;
000485         refnum : INTEGER;
000486         contiguous : BOOLEAN;
000487         count : LONGINT;
000488         VAR actual : LONGINT);
000489
000490     PROCEDURE TRUNCATE (VAR ecode : INTEGER;  refnum : INTEGER);
000491
000492     PROCEDURE COMPACT (VAR ecode : INTEGER;  refnum : INTEGER);
000493
000494     PROCEDURE RENAME_ENTRY ( VAR ecode:INTEGER; VAR path:pathname; VAR newname : e_name );
000495
000496     PROCEDURE READ_LABEL ( VAR ecode : INTEGER;
000497         VAR path : pathname;
000498         data_addr : LONGINT;
000499         count : LONGINT;
000500         VAR actual : LONGINT );
000501
000502     PROCEDURE WRITE_LABEL ( VAR ecode : INTEGER;
000503         VAR path : pathname;
000504         data_addr : LONGINT;
000505         count : LONGINT;
000506         VAR actual : LONGINT );
000507
000508     PROCEDURE MOUNT ( VAR ecode:INTEGER; VAR vname : e_name; VAR password : e_name ;
000509         VAR devname : e_name);
000510
000511     PROCEDURE UNMOUNT ( VAR ecode:INTEGER; VAR vname : e_name );
000512
000513     PROCEDURE SET_WORKING_DIR ( VAR ecode:INTEGER; VAR path:pathname );
000514
000515     PROCEDURE GET_WORKING_DIR ( VAR ecode:INTEGER; VAR path:pathname );
000516
000517     PROCEDURE SET_SAFETY ( VAR ecode:INTEGER; VAR path:pathname; on_off:BOOLEAN );
000518
000519     PROCEDURE DEVICE_CONTROL ( VAR ecode:INTEGER; VAR path:pathname;
000520         VAR cparm : dctype );
000521
000522     PROCEDURE RESET_CATALOG (VAR ecode : INTEGER; VAR path : pathname);
000523
```

Apple Lisa Computer Technical Information

```
000524 PROCEDURE RESET_SUBTREE ( VAR ecode : INTEGER;
000525                             VAR path  : pathname );
000526
000527 PROCEDURE GET_NEXT_ENTRY (VAR ecode : INTEGER; VAR prefix, entry : e_name);
000528
000529 PROCEDURE LOOKUP_NEXT_ENTRY ( VAR ecode   : INTEGER;
000530                             VAR prefix  : e_name;
000531                             VAR InfoRec : Q_Info );
000532
000533 PROCEDURE SET_FILE_INFO ( VAR ecode : INTEGER;
000534                          refnum : INTEGER;
000535                          fsi    : fs_info );
000536
000537
000538
000539
000540 { Process Management system calls }
000541
000542 FUNCTION My_ID : LONGINT;
000543
000544 PROCEDURE Info_Process (VAR errnum : INTEGER; proc_id : LONGINT;
000545                       VAR proc_info : procinfoRec);
000546
000547 PROCEDURE Yield_CPU (VAR errnum : INTEGER; to_any : BOOLEAN);
000548
000549 PROCEDURE SetPriority_Process (VAR errnum : INTEGER; proc_id : LONGINT;
000550                              new_priority : INTEGER);
000551
000552 PROCEDURE Suspend_Process (VAR errnum : INTEGER; proc_id : LONGINT;
000553                          susp_family : BOOLEAN);
000554
000555 PROCEDURE Activate_Process (VAR errnum : INTEGER; proc_id : LONGINT;
000556                          act_family : BOOLEAN);
000557
000558 PROCEDURE Kill_Process (VAR errnum : INTEGER; proc_id : LONGINT);
000559
000560 PROCEDURE Terminate_Process (VAR errnum : INTEGER; event_ptr : p_s_eventblk);
000561
000562 PROCEDURE Make_Process (VAR errnum : INTEGER; VAR proc_id : LONGINT;
000563                       VAR progfile : pathname; VAR entryname : namestring;
000564                       evt_chn_refnum : INTEGER);
000565
000566 PROCEDURE Sched_Class (VAR errnum : INTEGER; non_preempt : BOOLEAN);
000567
000568
000569
000570 { Memory Management system calls }
000571
000572 PROCEDURE make_dataseg(VAR errnum: INTEGER; VAR segname: pathname;
000573                      mem_size, disc_size: LONGINT; VAR refnum: INTEGER;
000574                      VAR segptr: LONGINT; ldsn: INTEGER; dstype: Tdstype);
000575
000576 PROCEDURE kill_dataseg (VAR errnum : INTEGER; VAR segname : pathname);
000577
000578 PROCEDURE open_dataseg (VAR errnum : INTEGER; VAR segname : pathname;
000579                       VAR refnum : INTEGER; VAR segptr : LONGINT;
000580                       ldsn : INTEGER);
000581
000582 PROCEDURE close_dataseg (VAR errnum : INTEGER; refnum : INTEGER);
000583
000584 PROCEDURE size_dataseg (VAR errnum : INTEGER; refnum : INTEGER;
000585                       deltamemsize : LONGINT; VAR newmemsize : LONGINT;
000586                       deltadiscsize: LONGINT; VAR newdiscsize: LONGINT);
000587
000588 PROCEDURE info_dataseg (VAR errnum : INTEGER; refnum : INTEGER;
000589                       VAR dsinfo : dsinfoRec);
```

Apple Lisa Computer Technical Information

```
000590
000591 PROCEDURE setaccess_dataseg (VAR errnum : INTEGER; refnum : INTEGER;
000592                             readonly : BOOLEAN);
000593
000594 PROCEDURE unbind_dataseg (VAR errnum : INTEGER; refnum : INTEGER);
000595
000596 PROCEDURE bind_dataseg(VAR errnum : INTEGER; refnum : INTEGER);
000597
000598 PROCEDURE info_ldsn (VAR errnum : INTEGER; ldsn: INTEGER; VAR refnum: INTEGER);
000599
000600 PROCEDURE flush_dataseg(VAR errnum: INTEGER; refnum: INTEGER);
000601
000602 PROCEDURE mem_info(VAR errnum: INTEGER;
000603                   VAR swapspace, dataspace,
000604                   cur_codesize, max_codesize: LONGINT);
000605
000606 PROCEDURE info_address(VAR errnum: INTEGER; address: LONGINT;
000607                       VAR refnum: INTEGER);
000608
000609
000610 { Exception Management system calls }
000611
000612
000613 PROCEDURE declare_excep_hdl (VAR errnum : INTEGER;
000614                             VAR excep_name : t_ex_name;
000615                             entry_point : longadr);
000616
000617 PROCEDURE disable_excep (VAR errnum : INTEGER;
000618                         VAR excep_name : t_ex_name;
000619                         queue : BOOLEAN);
000620
000621 PROCEDURE enable_excep (VAR errnum : INTEGER;
000622                       VAR excep_name : t_ex_name);
000623
000624 PROCEDURE signal_excep (VAR errnum : INTEGER;
000625                       VAR excep_name : t_ex_name;
000626                       excep_data : t_ex_data);
000627
000628 PROCEDURE info_excep (VAR errnum : INTEGER;
000629                     VAR excep_name : t_ex_name;
000630                     VAR excep_status : t_ex_sts);
000631
000632 PROCEDURE flush_excep (VAR errnum : INTEGER;
000633                      VAR excep_name : t_ex_name);
000634
000635
000636 { Event Channel management system calls }
000637
000638 PROCEDURE make_event_chn (VAR errnum : INTEGER;
000639                          VAR event_chn_name : pathname);
000640
000641 PROCEDURE kill_event_chn (VAR errnum : INTEGER;
000642                          VAR event_chn_name : pathname);
000643
000644 PROCEDURE open_event_chn (VAR errnum : INTEGER;
000645                          VAR event_chn_name : pathname;
000646                          VAR refnum : INTEGER;
000647                          VAR excep_name : t_ex_name;
000648                          receiver : BOOLEAN);
000649
000650 PROCEDURE close_event_chn (VAR errnum : INTEGER;
000651                          refnum : INTEGER);
000652
000653 PROCEDURE info_event_chn (VAR errnum : INTEGER;
000654                          refnum : INTEGER;
000655                          VAR chn_info : t_chn_sts);
```

Apple Lisa Computer Technical Information

```
000656
000657 PROCEDURE wait_event_chn (VAR errnum : INTEGER;
000658     VAR wait_list : t_waitlist;
000659     VAR refnum : INTEGER;
000660     event_ptr : p_r_eventblk);
000661
000662 PROCEDURE flush_event_chn (VAR errnum : INTEGER;
000663     refnum : INTEGER);
000664
000665 PROCEDURE send_event_chn (VAR errnum : INTEGER;
000666     refnum : INTEGER;
000667     event_ptr : p_s_eventblk;
000668     interval : timestmp_interval;
000669     clktime : time_rec);
000670
000671
000672 { Timer FUNCTIONS system calls }
000673
000674 PROCEDURE delay_time (VAR errnum : INTEGER;
000675     interval : timestmp_interval;
000676     clktime : time_rec);
000677
000678 PROCEDURE get_time (VAR errnum : INTEGER;
000679     VAR gmt_time : time_rec);
000680
000681 PROCEDURE convert_time (VAR errnum : INTEGER;
000682     VAR gmt_time : time_rec;
000683     VAR local_time : time_rec;
000684     to_gmt : BOOLEAN);
000685
000686
000687 {Configuration utilities: }
000688
000689 PROCEDURE CDINFO(VAR error: INTEGER;
000690     the_pos: cd_position;
000691     VAR the_info: cd_infobuf);
000692
000693 PROCEDURE CDKILL(VAR error: INTEGER;
000694     the_pos: cd_position);
000695
000696 PROCEDURE CDMAKE(VAR error: INTEGER;
000697     the_pos: cd_position;
000698     VAR this_info: cd_infobuf);
000699
000700 PROCEDURE MACH_INFO(VAR ecode : INTEGER; VAR the_info: minfo);
000701
000702 FUNCTION ENABLEDBG(enableit : BOOLEAN) : BOOLEAN;
000703
000704 PROCEDURE OSBOOTVOL(VAR error : INTEGER; VAR Volname: e_name);
000705
000706 PROCEDURE CARDS_EQUIPPED(VAR error : INTEGER;
000707     VAR in_slot : slot_array);
000708
000709 PROCEDURE Read_PMem (VAR errnum : INTEGER; VAR my_pMem : pMemRec );
000710
000711 PROCEDURE Write_PMem (VAR errnum : INTEGER; my_pMem : pMemRec);
000712
000713 PROCEDURE GetNxtConfig(VAR errnum: INTEGER; VAR NextEntry: LONGINT;
000714     VAR PMrec: PMemRec; VAR config: ConfigDev);
000715
000716 PROCEDURE PutNxtConfig(VAR errnum: INTEGER; VAR NextEntry: LONGINT;
000717     VAR PMrec: PMemRec; VAR config: ConfigDev);
000718
000719
000720 IMPLEMENTATION
000721
```

Apple Lisa Computer Technical Information

000722
000723 FINIS
000724

End of File -- Lines: 724 Characters: 26905

SUMMARY:

Total number of files : 1
Total file lines : 724
Total file characters : 26905