To Dave

ML System on Vax Unix

March 1982

1. Files Used for System Generation and Usage

The following files are used for system generation and usage.

```
LIBRARY      Makefile    array.ml   lambda.ml     ml.aloe    mlalloc.h  mlanal.h
mlanal.o     mlanal.p    mlassm.h   mlassm.o      mlassm.p   mlchanges.doc
mlcomp.h     mlcomp.o    mlcomp.p   mldebg.h      mldebg.o   mldebg.p   mlfcheck.c
mlfcheck.h   mlfcheck.o  mlglob.h   mlglob.o      mlkit.doc  mlmain.h
mlmain.o     mlmain.p    mlpars.h   mlpars.o      mlpars.p   mlrout.h   mlrout.o
mlrout.s     mlscan.h    mlscan.o   mlscan.p      mlserv.h   mlserv.o   mlserv.p
mlstor.h     mlstor.o    mlstor.p   mlsyntax.doc  mlsys      mlsys.mlisp
newmodel.ml  stree.ml
```

These files are classified as follows:

----.p ML system component modules written by Berkeley Pascal (system structure is most similar to the original one).

----.c file-id checking procedure written by C (to be called when the use command is executed).

----.h header files used when each of the modules is separately compiled (mlglob.h should be independently compiled because there are several global variable declarations in it).

----.s interpreter module written by Vax Unix assembly language.

----.o object code modules generated by Pascal compiler, C compiler and Vax Unix assembler.

----.ml several examples written in ML.

----.doc documentation files originally given by L. Cardelli.

mlsys executable code module of the ML system.

mlsys.mlisp macro definitions of emacs for editing.

LIBRARY library file originally given by L. Cardelli, and it is automatically read when mlsys starts.

Makefile makefile used by Unix make command to generate mlsys.

2. Notes for Using Mlsys

1. use command ---- The system checks the correctness of the file-id given as its parameter, and an error message comes out if something is wrong.

2. path name for Unix file ---- Since the character '/' is used as an escape character in mlsys, '//' should be typed in if you want to input '/'. In Unix file system, '/' is used as a separator for each directory. So, be careful when a full path name is given to specify a file. For example, //usr//ns//ml//newmodel.ml should be used instead of /usr/ns/ml/newmodel.ml . If one '/' is detected in a token, the system automatically print out '//' , and don't be confused.

3. Some bugs in garbage collection ---- There is some bugs in the garbage collection in this version.

3. How to Use the Editing Facility for Mlsys

Since the mlsys is running on Unix, it is possible to start it under Unix version emacs editor. There are three macros defined to combine the emacs to the mlsys.

1. call-mlsys (bound to ↑x↑m↑l)
   After invoking the emacs editor for any file of ML source, the mlsys can be started by ↑x↑m↑l key in. When the mlsys starts, the new window is created being bound to the "Shell" buffer and the screen is shared by these two windows. These windows and buffers can be handled by usual emacs command.

2. stat-transfer-beyond-buffer (bound to ↑x↑t)
   After invoking the mlsys, a user can edit any ML statement in the window bound to the file(buffer) first specified when starting the emacs. This macro transfers any one ML statement in this window to another bound to the Shell buffer and feeds it to the mlsys for its execution. This macro searches the ';' located at the end of a statement. It skips several ';''s used as separators in a list, record and variant. It also skips any comment line surrounded by '%'. When using this macro, the cursor(dot) should be set at any position between just after ';' of the end of the previous statement and the top of the statement a user wants to send to the mlsys.

3. statement-transfer-inside-buffer (bound to ↑x↑a)
   Since the "Shell' buffer is an emacs buffer, any editing command can be executed in this buffer. Any statement can be modified after its execution. This macro transfers a modified statement to the end of the buffer and feeds it to the mlsys for its execution. It also skips any ';' used as a separator in a list, record and variant, and skips any comment line. When using this macro, be careful to set the cursor(dot) just behind the prompt character '-' which the mlsys prints at the top of each statement. It is a user's responsibility to save the corrected statement by using any emacs command.

If there is a load command in your .emacs – pro file, these macros are automatically read at the emacs invocation.

4. Notes for Modifying Mlsys

Working spaces for runtime closures, function definitions and so forth are allocated by calling Unix system subroutine malloc. The sizes of these spaces are the same as the original ones. If a user needs more spaces for executing a large ML program, they can be extended by modifying the parameters for the malloc routine and some parts of the mlrout.s module. Since the Berkeley Vax Unix supports the virtual storage, the malloc allocates the virtual space. So the large size of the working space does not become the burden of the Unix Operating System.

Please contact to the following address if there is any comment or any problem:

Nobuo Saito
Dept. of Computer Science
Carnegie-Mellon University
Schenley Park
Pittsburgh, PA 15213

tel. (412) 578-3620