

File created: 19-Jan-94 13:35:27 {DSK}<sparky>export>users>nilsson>foreign-functions>FOREIGN-FUNCTION
S.;16

changes to: (IL:VARS IL:FOREIGN-FUNCTIONSCOMS)
(IL:STRUCTURES FOREIGN-POINTER)
(IL:SETFS ERROR-FLAG)
(IL:VARIABLES *ALL-FOREIGN-FUNCTIONS* *ALL-FOREIGN-FILES* VALID-C-TYPES *VALID-C-TYPES-MENU*)
(IL:FUNCTIONS C-FREE CHECK-FOREIGN-TYPE DEFFOREIGN DEF-C-STRUCT EXECUTABLE-P FOREIGN-ERROR-CASE
FOREIGN-FUNCTIONS-AROUNDEXITFN GET-FUNCTION GET-SYMBOL IL-TO-UNIX-FILENAME LINK-FILE MALLOC
UNLINK-FILE UNDEFINED-SYMBOLS SMASHING-APPLY ERROR-FLAG C-GETBASEBYTE GETBASEFLOAT
GETBASEINT GETBASEWORD GETBASEBYTE GETBASEBIT C-PUTBASEBYTE PUTBASEFLOAT PUTBASEINT
PUTBASEWORD PUTBASEBYTE PUTBASEBIT TRANSMOGRIFY-C-STRUCT)

previous date: 23-Dec-93 09:55:27 {DSK}<sparky>export>users>nilsson>foreign-functions>FOREIGN-FUNCTIONS.;15

Read Table: XCL

Package: FOREIGN-FUNCTIONS

Format: XCCS

; Copyright (c) 1992, 1993, 1994 by Venue. All rights reserved.

```
(IL:RPAQQ IL:FOREIGN-FUNCTIONSCOMS
((IL:ALISTS (IL:\INITSUBRS IL:CALL-C-FUNCTION IL:DLD-LINK IL:DLD-UNLINK-BY-FILE
            IL:DLD-UNLINK-BY-SYMBOL IL:DLD-GET-SYMBOL IL:DLD-GET-FUNC
            IL:DLD-FUNCTION-EXECUTABLE-P IL:DLD-LIST-UNDEFINED-SYMBOLS IL:C-MALLOC IL:C-FREE
            IL:C-PUTBASEBYTE IL:C-GETBASEBYTE IL:CALL-SMASHING-FUNCTION))
(IL:VARIABLES *ALL-FOREIGN-FUNCTIONS* *ALL-FOREIGN-FILES* VALID-C-TYPES *VALID-C-TYPES-MENU*
 *COFF-FILE-HEADER-SIZE* *AOUT-FILE-HEADER-SIZE* *FOREIGN-SYMBOLS*)
(IL:VARS ENCLOSING-TYPES)
(IL:FUNCTIONS C-FREE CHECK-FOREIGN-TYPE DEFFOREIGN DEF-C-STRUCT EXECUTABLE-P FOREIGN-ERROR-CASE
FOREIGN-FUNCTIONS-AROUNDEXITFN GET-FUNCTION GET-SYMBOL IL-TO-UNIX-FILENAME LINK-FILE MALLOC
UNLINK-FILE UNDEFINED-SYMBOLS)
;; Functions for Ron Kaplan's access mode.
(IL:FUNCTIONS SMASHING-APPLY ERROR-FLAG)
(IL:SETFS ERROR-FLAG)
;; Record defs.
(IL:FUNCTIONS TRANSMOGRIFY-C-STRUCT)
(IL:ADVARS (IL:CLISPRECORDTYPES C-STRUCT))
(IL:COMS                                     ; for handling datatype
 (IL:P (IL:MOVD 'IL:RECORD 'C-STRUCT)
 (IL:PUTPROP 'C-STRUCT 'IL:USERRECORDTYPE 'TRANSMOGRIFY-C-STRUCT)))
(IL:STRUCTURES FOREIGN-POINTER)
;; COFF stuff
(IL:RECORDS COFF-HEADER COFF-OPTIONAL-HEADER COFF-SECTION-HEADER)
(IL:FUNCTIONS READ-COFF-FILE)
;; AOUT stuff
(IL:RECORDS AOUT-HEADER AOUT-FILE N_LIST FOREIGN-SYMBOL-ENTRY)
(IL:FUNCTIONS READ-AOUT-HEADER REGISTER-AOUT-SYMBOLS N_TXTOFF N_DATOFF N_TREOFF N_DREOFF N_SYMOFF
 N_STROFF STRING-TABLE-SIZE GET-C-INTEGGER GET-C-SHORT GET-C-BYTE GET-C-ADDRESS)
(IL:P (PUSH 'FOREIGN-FUNCTIONS-AROUNDEXITFN IL:AROUNDEXITFNS))
(IL:PROP IL:MAKEFILE-ENVIRONMENT IL:FOREIGN-FUNCTIONS))

(IL:ADDTOVAR IL:\INITSUBRS
 (IL:CALL-C-FUNCTION 167)
 (IL:DLD-LINK 168)
 (IL:DLD-UNLINK-BY-FILE 169)
 (IL:DLD-UNLINK-BY-SYMBOL 170)
 (IL:DLD-GET-SYMBOL 171)
 (IL:DLD-GET-FUNC 172)
 (IL:DLD-FUNCTION-EXECUTABLE-P 173)
 (IL:DLD-LIST-UNDEFINED-SYMBOLS 174)
 (IL:C-MALLOC 175)
 (IL:C-FREE 176)
 (IL:C-PUTBASEBYTE 177)
 (IL:C-GETBASEBYTE 178)
 (IL:CALL-SMASHING-FUNCTION 179))

(DEFVAR *ALL-FOREIGN-FUNCTIONS* NIL
 "The list of all defined foreign functions on the form ((<name string> . <address>))*")

(DEFVAR *ALL-FOREIGN-FILES* NIL)

(DEFVAR VALID-C-TYPES)

(DEFVAR *VALID-C-TYPES-MENU* (IL:|create| IL:MENU
 IL:TITLE IL:_ "C types"
 IL:ITEMS IL:_ VALID-C-TYPES))
```

```
(DEFVAR *COFF-FILE-HEADER-SIZE* 20
"The size of the coff file header in bytes.")
```

```
(DEFVAR *AOUT-FILE-HEADER-SIZE* 32
"The size of the exec struct in bytes.")
```

```
(DEFVAR *FOREIGN-SYMBOLS* (MAKE-HASH-TABLE :TEST #'EQUAL)
"The global symbol table for the foreign symbols.")
```

```
(IL:RPAQQ ENCLOSING-TYPES (:CPOINTER :VECTOR :STRUCTURE))
```

```
(DEFUN C-FREE (POINTER SIZE)
(IL:SUBRCALL IL:C-FREE POINTER SIZE))
```

```
(DEFUN CHECK-FOREIGN-TYPE (TYPE &KEY VOID-ALLOWED-P)
(DECLARE (SPECIAL *VALID-C-TYPES-MENU*)
(LOOP (IF (IL:FMEMB TYPE VALID-C-TYPES)
(RETURN-FROM CHECK-FOREIGN-TYPE (CASE TYPE
(:VOID (IF VOID-ALLOWED-P
-1
(ERROR "Type :VOID is not allowed here.)))
(:INT (IL:\\TYPENUMBERFROMNAME 'IL:FIXP))
(:LONG (IL:\\TYPENUMBERFROMNAME 'IL:FIXP))
(:SHORT (IL:\\TYPENUMBERFROMNAME 'IL:FIXP))
(:CHAR (IL:\\TYPENUMBERFROMNAME 'IL:CHARACTER))
(:BYTE (IL:\\TYPENUMBERFROMNAME 'IL:FIXP))
(:LISPPTTR (IL:\\TYPENUMBERFROMNAME 'IL:FIXP))
(:CPOINTER (IL:\\TYPENUMBERFROMNAME 'IL:FIXP))
(:FLOAT (IL:\\TYPENUMBERFROMNAME 'IL:FLOATP))))
(RESTART-CASE (ERROR 'SIMPLE-ERROR :FORMAT-STRING "Bogus type for foreign function: ~s."
:FORMAT-ARGUMENTS (LIST TYPE))
(CONTINUE (NEW-TYPE)
:REPORT "Try new type." :INTERACTIVE (LAMBDA NIL (LIST (IL:MENU *VALID-C-TYPES-MENU*)
))
(SETQ TYPE NEW-TYPE))))))
```

```
(DEFMACRO DEFFOREIGN (FUNCTION (&REST ARGLIST)
&KEY RESULT-TYPE FOREIGN-NAME FUNCTION-DOCUMENTATION)
"Define a foreign function."
(SETQ FOREIGN-NAME (CTYPECASE FOREIGN-NAME (NULL (SYMBOL-NAME FUNCTION))
(STRING FOREIGN-NAME)))
(SETQ FUNCTION-DOCUMENTATION (AND (STRINGP FUNCTION-DOCUMENTATION)
FUNCTION-DOCUMENTATION))
(LET
((DESCRIPTOR-BLOCK (IL:\\ALLOCBLOCK (+ 5 (LENGTH ARGLIST))
NIL))
;; The conversion block looks like this:
; 1 function pointer.
; 2 RESULT-TYPE
; 3 ERRORFLAG
; 4 Number of args to the function.
; 5 0 if returnvalue on the stack else a pointer to a cell where the
; result should be stored. (This was ordered by Ron Kaplan /jarl)
; 6-... The argument types.
(FUNCARGS (IL:|for| ARG IL:|in| ARGLIST IL:|as| I IL:|from| 1 IL:|collect| (INTERN (IL:CONCAT "Arg-" I)
(SYMBOL-PACKAGE FUNCTION))))
(FUNCTION-POINTER (IL:SUBRCALL IL:DLD-GET-FUNC FOREIGN-NAME)))
(BLOCK
; If the function is on the *ALL-FOREIGN-FUNCTIONS* list then
; just stuff it there, else push the new def on the list.
CHECK-FUNCS
(DOLIST (A *ALL-FOREIGN-FUNCTIONS*)
(WHEN (EQUAL (CAR A)
FOREIGN-NAME)
(RPLACD A DESCRIPTOR-BLOCK)
(RETURN-FROM CHECK-FUNCS)))
(PUSH (CONS FOREIGN-NAME DESCRIPTOR-BLOCK)
*ALL-FOREIGN-FUNCTIONS*))
(IL:\\PUTBASEFIXP DESCRIPTOR-BLOCK 0
; If the function is defined and executable we set the 0'th position
; in DESCRIPTOR-BLOCK to the address, else the address is
; set to 0.
(IF (AND (< 16 FUNCTION-POINTER)
(EXECUTABLE-P FOREIGN-NAME))
FUNCTION-POINTER
0))
(IL:\\PUTBASEFIXP DESCRIPTOR-BLOCK 2
(CHECK-FOREIGN-TYPE RESULT-TYPE :VOID-ALLOWED-P T))
; Set the RESULT-TYPE
;; Leave a hole at 4 for the errorflag.
(IL:\\PUTBASEFIXP DESCRIPTOR-BLOCK 4 0)
(IL:\\PUTBASEFIXP DESCRIPTOR-BLOCK 6
; Set the # of args that we pass.
(LENGTH FUNCARGS))
;
```

;; Set smasher pointer to 0. That tells the emulator to return values instead of smashing them.

```
(IL:\PUTBASEFIXP DESCRIPTOR-BLOCK 8 0)
(DOTIMES (ARG# (LENGTH ARGLIST)) ; Set the typevector.
  (IL:\PUTBASEFIXP DESCRIPTOR-BLOCK (+ 10 (* 2 ARG#))
    (CHECK-FOREIGN-TYPE (NTH ARG# ARGLIST)
      :VOID-ALLOWED-P NIL)))
(SETF (GET FUNCTION 'FOREIGN-NAME) ; Keep name and descriptorblock around.
  FOREIGN-NAME)
(SETF (GET FUNCTION 'DESCRIPTOR-BLOCK)
  DESCRIPTOR-BLOCK)
(EVAL
  \ (DEFUN ,FUNCTION ,FUNCARGS
    ,@FUNCTION-DOCUMENTATION
    (LET ((RESULT (IL:SUBRCALL IL:CALL-C-FUNCTION ,DESCRIPTOR-BLOCK ,@FUNCARGS))
      (ERRNO (IL:\GETBASEFIXP ,DESCRIPTOR-BLOCK 4)))
      (CASE ERRNO
        (0 T)
        (-1 (ERROR "Foreign function ~s is not executable." ,FOREIGN-NAME))
        (-2 (ERROR "Bogus return type."))
        (T , (WHEN FUNCARGS
          \ (ERROR "Type of argument# ~d (~s) is not ~s as declared." ERRNO
            (TYPE-OF (NTH ERRNO (LIST ,@FUNCARGS)))
            (IL:[fetch] IL:DTDNAME IL:[of] (IL:\GETDTD (IL:NTPX (IL:\GETBASEFIXP
              ,DESCRIPTOR-BLOCK
              (+ 8 (* 2 ERRNO))))))))))
        , (IF (EQUAL RESULT-TYPE :VOID)
          ' (VALUES) ; If the result type is :VOID it is only fair that we return (VALUES)
          ' RESULT ; ELSE let the emulator take care of the type conversion.
        )
      )
    )
  )
(SETF (GET 'IL:\GETBASEFIXP 'COMPILER::SIDE-EFFECTS-DATA)
  NIL)
(COMPILE FUNCTION)
(SETF (GET 'IL:\GETBASEFIXP 'COMPILER::SIDE-EFFECTS-DATA)
  ' (:NONE . :NONE))
(LIST 'QUOTE FUNCTION)))
```

(DEFMACRO DEF-C-STRUCT (FOOT 42)

```
(DEFUN EXECUTABLE-P (NAME)
  (DECLARE (TYPE (OR STRING SYMBOL)
    NAME))
  (LET* ((NAME (CTYPECASE NAME (SYMBOL (OR ; See if we stored the name.
    (GET NAME 'FOREIGN-NAME)
    ; If not, try the symbol name.
    (SYMBOL-NAME NAME)))
    (STRING NAME)))
    (RESULT (IL:SUBRCALL IL:DLD-FUNCTION-EXECUTABLE-P NAME)))
    (IF (ZEROP RESULT)
      NIL
      T)))
```

(DEFUN FOREIGN-ERROR-CASE (DLD-ERROR-NUMBER)

```
(CASE DLD-ERROR-NUMBER
  (1 "Can't open foreign file ~s.")
  (2 "Bad magic number in foreign file ~s")
  (3 "Faiiliure reading header in foreign file ~s")
  (4 "Premature EOF in text section of foreign file ~s")
  (5 "Premature EOF in symbol section of foreign file ~s")
  (6 "Bad string table in foreign file ~s")
  (7 "Premature EOF in text relocation of foreign file ~s")
  (8 "Premature EOF in data section in foreign file ~s")
  (9 "Premature EOF in data relocation in foreign file ~s")
  (10 "Multiple definitions of symbol in foreign file ~s")
  (11 "Malformed library archive (foreign file ~s)")
  (12 "Common block not supported (foreign file ~s)")
  (13 "Malformed input file (foreign file ~s)")
  (14 "Bad relocation info (foreign file ~s)")
  (15 "Virtual memory exhausted while loading foreign file ~s.")
  (16 "Undefined symbol in foreign file ~s.")
  (T (CERROR "CONTINUE?" "BOGUS ERROR CODE IN DLD.))))
```

(DEFUN FOREIGN-FUNCTIONS-AROUNDXITFN (EVENT)

```
(CASE EVENT
  ((IL:AFTERLOGOUT IL:AFTERMAKESYS IL:AFTERSAVEVM IL:AFTERSYSOUT)
    (DOLIST (F *ALL-FOREIGN-FILES*) ; Attempt to link the files we had in memory.
      (LINK-FILE F))
    (DOLIST (A *ALL-FOREIGN-FUNCTIONS* ; Redefine the functions.
      A)
      (LET ((FUNCTION-POINTER (IL:SUBRCALL IL:DLD-GET-FUNC (CAR A)))
        (IL:\PUTBASEFIXP (CDR A)
          0
```



```

                                (LIST PATHNAME))
                                (CONTINUE (NEW-PATHNAME)
                                :REPORT "Try another file." :INTERACTIVE
                                (LAMBDA NIL (LIST (IL:PROMPTFORWORD "New file name:" (NAMESTRING
                                PATHNAME))))
                                (SETQ PATHNAME NEW-PATHNAME))))))
;; Run down the list of defined functions and see if we can resolve any references.
(PUSH PATHNAME *ALL-FOREIGN-FILES*) ; Remember this file for later.
(DOLIST (A *ALL-FOREIGN-FUNCTIONS* ; car is the name cdr is the descriptor.
)
)
(WHEN (ZEROP (IL:\GETBASE (CDR A)
1))
(LET ((FUNCTION-POINTER (IL:SUBRCALL IL:DLD-GET-FUNC (CAR A)))
(IL:\PUTBASEFIXP (CDR A)
0
(IF (AND (< 16 FUNCTION-POINTER)
(EXECUTABLE-P (CAR A))
FUNCTION-POINTER
0))))))
(DEFUN MALLOC (SIZE)
(IL:SUBRCALL IL:C-MALLOC SIZE))
(DEFUN UNLINK-FILE (NAME &KEY (SYMBOL-NAME-P NIL)
(FORCE-P NIL))
;; Do the raw unlinking.
(PROG1 (BLOCK GUARD
(LOOP (LET ((NAME (IL-TO-UNIX-FILENAME (SYMBOL-NAME (IL:FINDFILE (CTYPEPCASE NAME
(SYMBOL (SYMBOL-NAME NAME))
(STRING NAME)
(PATHNAME (NAMESTRING NAME)
))))))
(RESULT (IF SYMBOL-NAME-P
(IL:SUBRCALL IL:DLD-UNLINK-BY-SYMBOL NAME (IF FORCE-P
1
0))
(IL:SUBRCALL IL:DLD-UNLINK-BY-FILE NAME (IF FORCE-P
1
0))))))
(IF (ZEROP RESULT)
(RETURN-FROM GUARD NAME)
(RESTART-CASE (ERROR 'SIMPLE-ERROR :FORMAT-STRING (DLD-ERROR-CASE RESULT)
:FORMAT-ARGUMENTS
(LIST NAME))
(CONTINUE (NEW-NAME)
:REPORT "Try another foreign symbol." :INTERACTIVE
(LAMBDA NIL (LIST (IL:PROMPTFORWORD "New foreign name:" NAME)))
(SETQ NAME NEW-NAME))))))
(SETQ *ALL-FOREIGN-FILES* ; Forget that this file was loaded.
(REMOVE NAME *ALL-FOREIGN-FILES*))
;; Run down the list of defined functions and revalidate them.
(DOLIST (A *ALL-FOREIGN-FUNCTIONS* ; car is the name cdr is the descriptor.
)
)
(WHEN (OR (< 16 (IL:SUBRCALL IL:DLD-GET-FUNC (CAR A))
(NOT (EXECUTABLE-P (CAR A))))
(IL:\PUTBASEFIXP (CDR A)
0 0))))))
(DEFUN UNDEFINED-SYMBOLS ()
(LET ((HEADPOINTER ; This is a pointer to an array of pointers to a string
(IL:SUBRCALL IL:DLD-LIST-UNDEFINED-SYMBOLS))
S)
(WHEN HEADPOINTER
(DOTIMES (OFFSET (C-GETBASEBYTE
;; Number of undefined symbols.
(GET-SYMBOL "dld_undefined_sym_count")
0 :INT))
(LET ((STRINGPOINTER (C-GETBASEBYTE HEADPOINTER OFFSET :INT)))
(DO* ((CHARPTR 1 ; Start at index 1 to avoid leading #\_ in the name
(1+ CHARPTR))
(CHAR (CHARACTER (C-GETBASEBYTE STRINGPOINTER CHARPTR :BYTE))
(CHARACTER (C-GETBASEBYTE STRINGPOINTER CHARPTR :BYTE)))
(STRN (LIST CHAR)
(CONS CHAR STRN)))
((EQL CHAR #\Null)
(PUSH (MAP 'STRING #'IDENTITY (REVERSE
; STRN is in reverse order
(CDR STRN)))
; Get rid of the #\Null
S

```

```
))))))
S))
```

:: Functions for Ron Kaplan's access mode.

```
(DEFMACRO SMASHING-APPLY (DESCRIPTOR PLACE &REST ARGS)
  `(IL:SUBRCALL IL:CALL-SMASHING-FUNCTION ,DESCRIPTOR ,PLACE ,@ARGS))
```

```
(DEFMACRO ERROR-FLAG (DESCRIPTOR)
  `(IL:\\GETBASEFIXP ,DESCRIPTOR 4))
```

```
(DEFSETF ERROR-FLAG (DESCRIPTOR) (NEWVAL)
  `(IL:\\PUTBASEFIXP ,DESCRIPTOR 4 ,NEWVAL))
```

:: Record defs.

```
(DEFUN TRANSMOGRIFY-C-STRUCT (STRUCTURE-DESCRIPTION)
```

:: Test the description for discrepancies and build a description of the slots.

```
(LET ((NAME (SECOND STRUCTURE-DESCRIPTION))
      (BODY (THIRD STRUCTURE-DESCRIPTION))
      (DESCRIPTOR NIL)
      (BYTE-ADDR 0)
      (LST NIL))
```

:: The format of a field is (FIELDNAME TYPE <typemodifier>) where the modifier is either :POINTER :STRUCTURE or an integer denoting that it is an array.

```
(MACROLET ((MAKE-ACCESSOR (D GET PUT OFFSET)
  `(, (FIRST D)
    (, GET 'IL:DATUM ,OFFSET)
    (, PUT 'IL:DATUM ,OFFSET IL:NEWVALUE))))
  (DOLIST (D BODY)
    (LET ((BASE BYTE-ADDR)
          (CASE (SECOND D)
            (:BIT (INCF BYTE-ADDR)
              :: 8 bit addr. No address adjustment.
            (:CHAR
              (PUSH (MAKE-ACCESSOR D GETBASEBYTE PUTBASEBYTE BYTE-ADDR)
                LST)
              (INCF BYTE-ADDR))
            (:BYTE
              (PUSH (MAKE-ACCESSOR D GETBASEBYTE PUTBASEBYTE BYTE-ADDR)
                LST)
              (INCF BYTE-ADDR))
              :: 16 bit addr. Adjust address to even boundaries.
            (:SHORT
              (WHEN (ODDP BYTE-ADDR)
                (INCF BYTE-ADDR))
              (PUSH (MAKE-ACCESSOR D GETBASEWORD PUTBASEWORD (ASH BYTE-ADDR -1))
                LST)
              (INCF BYTE-ADDR 2))
              :: 32 bit addr. Adjust address to 4 boundaries.
            (:INT
              (INCF BYTE-ADDR (MOD (- 4 (MOD BYTE-ADDR 4))
                4))
              (PUSH (MAKE-ACCESSOR D GETBASEINT PUTBASEINT (ASH BYTE-ADDR -2))
                LST)
              (INCF BYTE-ADDR 4))
            (:LONG
              (INCF BYTE-ADDR (MOD (- 4 (MOD BYTE-ADDR 4))
                4))
              (PUSH (MAKE-ACCESSOR D GETBASEINT PUTBASEINT (ASH BYTE-ADDR -2))
                LST)
              (INCF BYTE-ADDR 4))
            (:FLOAT
              (INCF BYTE-ADDR (MOD (- 4 (MOD BYTE-ADDR 4))
                4))
              (PUSH (MAKE-ACCESSOR D GETBASEFLOAT PUTBASEFLOAT (ASH BYTE-ADDR -2))
                LST)
              (INCF BYTE-ADDR 4))))))
    `(IL:ACCESSFNS ,NAME ,(REVERSE LST)
      (CREATE (IL:\\\\ALLOCBLOCK (ASH BYTE-ADDR -2))))))
```

```
(IL:ADDTOVAR IL:CLISPRECORDTYPES C-STRUCT)
```

:: for handling datatype

```
(IL:MOVD 'IL:RECORD 'C-STRUCT)
```

(IL:PUTPROP 'C-STRUCT 'IL:USERRECORDTYPE 'TRANSMOGRIFY-C-STRUCT)

(DEFSTRUCT FOREIGN-POINTER
"Pointer to a foreign object"
(DESTINATION-TYPE NIL)
(VALUE NIL))

:: COFF stuff

(IL:DECLARE\ : IL:EVAL@COMPILE

(IL:BLOCKRECORD COFF-HEADER ((F_MAGIC
IL:BITS 16)
(F_NSCNS
IL:BITS 16)
(F_TIMDAT
IL:BITS 32)
(F_SYMPTR
IL:BITS 32)
(F_NSYSMS
IL:BITS 32)
(F_OPTHEADER
IL:BITS 16)
(F_FLAGS
IL:BITS 16)))

(IL:BLOCKRECORD COFF-OPTIONAL-HEADER ((MAGIC IL:BITS 16)
(VSTAMP IL:BITS 16)
(TSIZE IL:BITS 32)
(DSIZE IL:BITS 32)
(BSIZE IL:BITS 32)
(ENTRY IL:BITS 32)
(TEXT_START
IL:BITS 32)
(DATA_START
IL:BITS 32)))

(IL:BLOCKRECORD COFF-SECTION-HEADER ((S_NAME1
IL:BITS 32)
(S_NAME2
IL:BITS 32)
(S_PADDR
IL:BITS 32)
(S_VADDR
IL:BITS 32)
(S_SIZE
IL:BITS 32)
(S_SCNPTR
IL:BITS 32)
(S_RELPTR
IL:BITS 32)
(S_LNNOPTR
IL:BITS 32)
(S_NRELOC
IL:BITS 16)
(S_NLNNO
IL:BITS 16)
(S_FLAGS
IL:BITS 32)))

(DEFUN READ-COFF-FILE (FILENAME)

(LET* ((FILEHEADER (MAKE-ARRAY *COFF-FILE-HEADER-SIZE* :ELEMENT-TYPE '(UNSIGNED-BYTE 8)
:ADJUSTABLE NIL))
(FILEHEADERBASE (IL:|fetch| (IL:ONED-ARRAY IL:BASE) IL:|of| FILEHEADER))
(OPTIONALHEADER (MAKE-ARRAY '(100)
:ELEMENT-TYPE
'(UNSIGNED-BYTE 8)
:ADJUSTABLE NIL))
(OPTHEADERBASE (IL:|fetch| (IL:ONED-ARRAY IL:BASE) IL:|of| OPTIONALHEADER)))
(WITH-OPEN-FILE (FILE FILENAME :IF-DOES-NOT-EXITS :ERROR :ELEMENT-TYPE '(UNSIGNED-BYTE 8)
:DIRECTION :INPUT)
(DOTIMES (INDEX *COFF-FILE-HEADER-SIZE*)
(SETF (AREF FILEHEADER INDEX)
(READ-BYTE FILE :EOF-ERROR-P T)))
(FORMAT T "opthead size: ~d~%" (IL:|fetch| (COFF-HEADER F_OPTHEADER) IL:|of| FILEHEADERBASE))
(IL:|if| (PLUSP (IL:|fetch| (COFF-HEADER F_OPTHEADER) IL:|of| FILEHEADERBASE))
IL:|then| (DOTIMES (INDEX (IL:|fetch| (COFF-HEADER F_OPTHEADER) IL:|of| FILEHEADERBASE))
(SETF (AREF OPTIONALHEADER INDEX)
(READ-BYTE FILE :EOF-ERROR-P T)))
(FORMAT T "Magic: ~o~%" (IL:|fetch| (COFF-OPTIONAL-HEADER MAGIC) IL:|of| OPTHEADERBASE))
(FORMAT T "Text size: ~d~%" (IL:|fetch| (COFF-OPTIONAL-HEADER TSIZE) IL:|of| OPTHEADERBASE))
(FORMAT T "data size: ~d~%" (IL:|fetch| (COFF-OPTIONAL-HEADER DSIZE) IL:|of| OPTHEADERBASE))
(FORMAT T "unit data size: ~d~%" (IL:|fetch| (COFF-OPTIONAL-HEADER BSIZE) IL:|of|
OPTHEADERBASE

```

    )))
    (FORMAT T "Number of symtab entries: ~b~&" (IL:|fetch| (COFF-HEADER F_NSYMS) IL:|of| FILEHEADERBASE))
))

;; AOUT stuff

(IL:DECLARE\ : IL:EVAL@COMPILE

(IL:BLOCKRECORD AOUT-HEADER ( (A_MAGIC
    IL:BITS 32)
    (A_TEXT
    IL:BITS 32)
    (A_DATA
    IL:BITS 32)
    (A_BSS
    IL:BITS 32)
    (A_SYMS
    IL:BITS 32)
    (A_ENTRY
    IL:BITS 32)
    (A_TRSIZE
    IL:BITS 32)
    (A_DRSIZE
    IL:BITS 32)))

(IL:DATATYPE AOUT-FILE (NAME HEADER TEXT DATA TEXT-RELOC DATA-RELOC SYMBOL-TABLE STRING-TABLE))

(IL:BLOCKRECORD N_LIST
    ( (N_NAME
    IL:BITS 32)
    (N_MISC
    IL:BITS 32)
    (N_VALUE
    IL:BITS 32)))

(IL:DATATYPE FOREIGN-SYMBOL-ENTRY (NAME TYPE EXTERNAL-P VALUE-INDEX OBJECTFILE)
    (IL:ACCESSFNS (VALUE (IL:|with| FOREIGN-SYMBOL-ENTRY IL:DATUM (CASE TYPE
        (:UNDEFINED :UNDEFINED)
        (:ABSOLUTE )
        (:TEXT )
        (:DATA (GET-C-INTEGGER (IL:|fetch|
            (AOUT-FILE HEADER)
            IL:|of| OBJECTFILE)
            VALUE-INDEX))
        (:BSS )
        (:COMMON )
        (:FILE-NAME ))))))))

)

(IL:/DECLAREDATATYPE 'AOUT-FILE ' (IL:POINTER IL:POINTER IL:POINTER IL:POINTER IL:POINTER IL:POINTER IL:POINTER
    IL:POINTER)

    ;; ---field descriptor list elided by lister---
    ' 16)

(IL:/DECLAREDATATYPE 'FOREIGN-SYMBOL-ENTRY ' (IL:POINTER IL:POINTER IL:POINTER IL:POINTER IL:POINTER)

    ;; ---field descriptor list elided by lister---
    ' 10)

(DEFUN READ-AOUT-HEADER (FILENAME)
    (WITH-OPEN-FILE (FILE FILENAME :IF-DOES-NOT-EXITS :ERROR :ELEMENT-TYPE '(UNSIGNED-BYTE 8)
        :DIRECTION :INPUT)
        (LET* ((OBJECTARRAY (MAKE-ARRAY (FILE-LENGTH FILE)
            :ELEMENT-TYPE
            '(UNSIGNED-BYTE 8)
            :ADJUSTABLE NIL))
            (OBJECTBASE (IL:|fetch| (IL:ONED-ARRAY IL:BASE) IL:|of| OBJECTARRAY))
            (AOUTSTRUCTURE NIL))
            (DOTIMES (INDEX (FILE-LENGTH FILE))
                (SETF (AREF OBJECTARRAY INDEX)
                    (READ-BYTE FILE :EOF-ERROR-P T)))
            (SETQ AOUTSTRUCTURE (IL:|create| AOUT-FILE
                NAME IL:_ FILENAME
                ;; Header is the start of the whole array,
                HEADER IL:_ OBJECTARRAY
                ;; Text is the start of the code array
                TEXT IL:_ (MAKE-ARRAY (LIST (IL:|fetch| (AOUT-HEADER A_TEXT)
                    IL:|of| OBJECTBASE))
                    :ELEMENT-TYPE
                    '(UNSIGNED-BYTE 8)
                    :DISPLACED-TO OBJECTARRAY :DISPLACED-INDEX-OFFSET
                    (N_TXTOFF
                    OBJECTARRAY))

```



```

;; DATA start = aout-end-index + textsize
DATA IL:_ (MAKE-ARRAY (LIST (IL:|fetch| (AOUT-HEADER A_DATA)
                             IL:|of| OBJECTBASE))
                    :ELEMENT-TYPE
                    '(UNSIGNED-BYTE 8)
                    :DISPLACED-TO OBJECTARRAY :DISPLACED-INDEX-OFFSET
                    (N_DATOFF
                     OBJECTARRAY))
TEXT-RELOC IL:_ (MAKE-ARRAY (IL:|fetch| (AOUT-HEADER A_TRSIZE)
                                       IL:|of| OBJECTBASE)
                          :ELEMENT-TYPE
                          '(UNSIGNED-BYTE 8)
                          :DISPLACED-TO OBJECTARRAY
                          :DISPLACED-INDEX-OFFSET (N_TRELOFF
                                                    OBJECTARRAY))
DATA-RELOC IL:_ (MAKE-ARRAY (IL:|fetch| (AOUT-HEADER A_DRSIZE)
                                       IL:|of| OBJECTBASE)
                          :ELEMENT-TYPE
                          '(UNSIGNED-BYTE 8)
                          :DISPLACED-TO OBJECTARRAY
                          :DISPLACED-INDEX-OFFSET (N_DRELOFF
                                                    OBJECTARRAY))
SYMBOL-TABLE IL:_ (MAKE-ARRAY (LIST (IL:|fetch| (AOUT-HEADER A_SYMS)
                                              IL:|of| OBJECTBASE))
                             :ELEMENT-TYPE
                             '(UNSIGNED-BYTE 8)
                             :DISPLACED-TO OBJECTARRAY
                             :DISPLACED-INDEX-OFFSET (N_SYMOFF
                                                       OBJECTARRAY))
STRING-TABLE IL:_ (MAKE-ARRAY (LIST (STRING-TABLE-SIZE OBJECTARRAY))
                             :ELEMENT-TYPE
                             '(UNSIGNED-BYTE 8)
                             :DISPLACED-TO OBJECTARRAY
                             :DISPLACED-INDEX-OFFSET (N_STROFF
                                                       OBJECTARRAY)))

```

;; Make Medley believe that this is an array of string-char instead. This is ugly but it works. /Jarl.

```

(IL:|replace| (IL:ONED-ARRAY IL:TYPE-NUMBER) IL:|of| (IL:|fetch| (AOUT-FILE STRING-TABLE) IL:|of|
                                                    AOUTSTRUCTURE
                                                    )
            (IL:|with| 67)
AOUTSTRUCTURE)))

```

```

(DEFUN REGISTER-AOUT-SYMBOLS (AOUFILERECOND)
  (LET ((SYMBOL-TABLE (IL:|fetch| (AOUT-FILE SYMBOL-TABLE) IL:|of| AOUFILERECOND))
        (STRING-TABLE (IL:|fetch| (AOUT-FILE STRING-TABLE) IL:|of| AOUFILERECOND)))
    (DO ((RECORDINDEX 0 (+ RECORDINDEX 12)))
        ((>= RECORDINDEX (LENGTH SYMBOL-TABLE)))
      (LET* ((STRINGTAB-INDEX (GET-C-INTEGGER SYMBOL-TABLE RECORDINDEX))
             (TYPE-ENTRY (GET-C-BYTE SYMBOL-TABLE (+ 4 RECORDINDEX)))
             (OTHER-ENTRY (GET-C-BYTE SYMBOL-TABLE (+ 5 RECORDINDEX)))
             (DESCRIPTION (GET-C-SHORT SYMBOL-TABLE (+ 6 RECORDINDEX)))
             (VALUE-INDEX (GET-C-INTEGGER SYMBOL-TABLE (+ 8 RECORDINDEX)))
             (NAME (STRING (SUBSEQ STRING-TABLE STRINGTAB-INDEX (POSITION #\Null STRING-TABLE :START
                                                                              STRINGTAB-INDEX))))
             (REC (IL:|create| FOREIGN-SYMBOL-ENTRY
                            NAME IL:_ NAME
                            OBJECTFILE IL:_ AOUFILERECOND
                            EXTERNAL-P IL:_ (ODDP TYPE-ENTRY)
                            TYPE IL:_ (CASE (LOGAND TYPE-ENTRY 30)
                                             (0 :UNDEFINED)
                                             (2 :ABSOLUTE)
                                             (4 :TEXT)
                                             (6 :DATA)
                                             (8 :BSS)
                                             (18 :COMMON)
                                             (30 :FILE-NAME))))))
          (SETF (GETHASH NAME *FOREIGN-SYMBOLS*)
                REC)
          (CASE (IL:|fetch| (FOREIGN-SYMBOL-ENTRY TYPE) IL:|of| REC)
                (:UNDEFINED )
                (:ABSOLUTE )
                (:TEXT )
                (:DATA (IL:|replace| (FOREIGN-SYMBOL-ENTRY VALUE-INDEX) IL:|of| REC IL:|with| (+ VALUE-INDEX
                                                                                               *AOUT-FILE-HEADER-SIZE*
                                                                                               )))
                (:BSS )
                (:COMMON )
                (:FILE-NAME ))
          REC))))

```

```

(DEFUN N_TXTOFF (OBJECT)
  *AOUT-FILE-HEADER-SIZE*)

```

```

(DEFUN N DATOFF (OBJECTARRAY)
  (+ (N_TXTOFF
      OBJECTARRAY)
      (IL:|fetch| (AOUT-HEADER A_TEXT) IL:|of| (IL:|fetch| (IL:ONED-ARRAY IL:BASE) IL:|of| OBJECTARRAY))))

```

```

(DEFUN N TRELOFF (OBJECTARRAY)
  (+ (N_DATOFF
      OBJECTARRAY)
      (IL:|fetch| (AOUT-HEADER A_DATA) IL:|of| (IL:|fetch| (IL:ONED-ARRAY IL:BASE) IL:|of| OBJECTARRAY))))

```

```

(DEFUN N DRELOFF (OBJECTARRAY)
  (+ (N_TRELOFF
      OBJECTARRAY)
      (IL:|fetch| (AOUT-HEADER A_TRSIZE) IL:|of| (IL:|fetch| (IL:ONED-ARRAY IL:BASE) IL:|of| OBJECTARRAY))))

```

```

(DEFUN N SYMOFF (OBJECTARRAY)
  (+ (N_DRELOFF
      OBJECTARRAY)
      (IL:|fetch| (AOUT-HEADER A_DRSIZE) IL:|of| (IL:|fetch| (IL:ONED-ARRAY IL:BASE) IL:|of| OBJECTARRAY))))

```

```

(DEFUN N STROFF (OBJECTARRAY)
  (+ (N_SYMOFF
      OBJECTARRAY)
      (IL:|fetch| (AOUT-HEADER A_SYMS) IL:|of| (IL:|fetch| (IL:ONED-ARRAY IL:BASE) IL:|of| OBJECTARRAY))))

```

```

(DEFUN STRING-TABLE-SIZE (OBJECTARRAY)
  (LET* ((INDEX (N_STROFF
                  OBJECTARRAY))
         (RESULT (IL:\GETBASEBYTE (IL:|fetch| (IL:ONED-ARRAY IL:BASE) IL:|of| OBJECTARRAY)
                                   INDEX)))
    (DOTIMES (A 3)
      (SETQ RESULT (+ (IL:LSH RESULT 8)
                     (IL:\GETBASEBYTE (IL:|fetch| (IL:ONED-ARRAY IL:BASE) IL:|of| OBJECTARRAY)
                                       (INCF INDEX))))))
    RESULT))

```

```

(DEFUN GET-C-INTEGGER (ARRAY INDEX)
  (+ (IL:LSH (AREF ARRAY INDEX)
            24)
      (IL:LSH (AREF ARRAY (+ INDEX 1))
            16)
      (IL:LSH (AREF ARRAY (+ INDEX 2))
            8)
      (AREF ARRAY (+ INDEX 3))))

```

```

(DEFUN GET-C-SHORT (ARRAY INDEX)
  (+ (IL:LSH (AREF ARRAY INDEX)
            8)
      (AREF ARRAY (+ INDEX 1))))

```

```

(DEFUN GET-C-BYTE (ARRAY INDEX)
  (AREF ARRAY INDEX))

```

```

(DEFUN GET-C-ADDRESS ()
  (ERROR "NOT YET!"))

```

```

(PUSH 'FOREIGN-FUNCTIONS-AROUNDEXITFN IL:AROUNDEXITFNS)

```

```

(IL:PUTPROPS IL:FOREIGN-FUNCTIONS IL:MAKEFILE-ENVIRONMENT (:READTABLE "XCL" :PACKAGE
                                                           (XCL:DEFPACKAGE "FOREIGN-FUNCTIONS"
                                                           (:USE "CL" "CONDITIONS")
                                                           (:NICKNAMES "FF")
                                                           (:EXPORT "DEFFOREIGN" "DEF-C-STRUCT"
                                                           "MALLOC" "C-FREE"
                                                           "C-GETBASEBYTE" "GETBASEFLOAT"
                                                           "GETBASEINT" "GETBASEWORD"
                                                           "GETBASEBYTE" "GETBASEBIT"
                                                           "LINK-FILE" "UNLINK-FILE"
                                                           "UNDEFINED-SYMBOLS"
                                                           "EXECUTABLE-P" "C-PUTBASEBYTE"
                                                           "PUTBASEFLOAT" "PUTBASEINT"
                                                           "PUTBASEWORD" "PUTBASEBYTE"
                                                           "PUTBASEBIT")))
          :BASE 10))

```

```

(IL:PUTPROPS IL:FOREIGN-FUNCTIONS IL:COPYRIGHT ("Venue" 1992 1993 1994))

```

FUNCTION INDEX

C-FREE	2	GET-FUNCTION	4	N_TRELOFF	10
CHECK-FOREIGN-TYPE	2	GET-SYMBOL	4	N_TXTOFF	9
EXECUTABLE-P	3	IL-TO-UNIX-FILENAME	4	READ-AOUT-HEADER	8
FOREIGN-ERROR-CASE	3	LINK-FILE	4	READ-COFF-FILE	7
FOREIGN-FUNCTIONS-AROUNDEXITFN	3	MALLOC	5	REGISTER-AOUT-SYMBOLS	9
GET-C-ADDRESS	10	N_DATOFF	10	STRING-TABLE-SIZE	10
GET-C-BYTE	10	N_DRELOFF	10	TRANSMOGRIFY-C-STRUCT	6
GET-C-INTEGGER	10	N_STROFF	10	UNDEFINED-SYMBOLS	5
GET-C-SHORT	10	N_SYMOFF	10	UNLINK-FILE	5

VARIABLE INDEX

ALL-FOREIGN-FILES	1	*COFF-FILE-HEADER-SIZE*	2	IL:CLISPRECORDTYPES	6	IL:\\INITSUBRS	1
ALL-FOREIGN-FUNCTIONS	1	*FOREIGN-SYMBOLS*	2	ENCLOSING-TYPES	2		
AOUT-FILE-HEADER-SIZE	2	*VALID-C-TYPES-MENU*	1	VALID-C-TYPES	1		

RECORD INDEX

AOUT-FILE	8	COFF-HEADER	7	COFF-SECTION-HEADER	7	N_LIST	8
AOUT-HEADER	8	COFF-OPTIONAL-HEADER	7	FOREIGN-SYMBOL-ENTRY	8		

MACRO INDEX

DEF-C-STRUCT	3	DEFFOREIGN	2	ERROR-FLAG	6	SMASHING-APPLY	6
--------------------	---	------------------	---	------------------	---	----------------------	---

PROPERTY INDEX

IL:FOREIGN-FUNCTIONS	10
----------------------------	----

STRUCTURE INDEX

FOREIGN-POINTER	7
-----------------------	---

SETF INDEX

ERROR-FLAG	6
------------------	---
