

File created: 5-Nov-92 15:57:00 {Pele:mv:envos}<LispCore>Sources>MEM.;7

changes to: (FNS MKI.NEXTPAGE MKI.NEWPAGE WriteoutPage I.PUTBASEPTR I.GETBASEFIXP I.PUTBASEFIXP)
(VARS MEMCOMS)

previous date: 16-May-90 20:36:32 {Pele:mv:envos}<LispCore>Sources>MEM.;2

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

;;
;; Copyright (c) 1982, 1983, 1986, 1990, 1992 by Venue & Xerox Corporation. All rights reserved.

(RPAQQ **MEMCOMS**

;; Memory-faking functions for use in MAKEINIT. These read and write from a huge (tree-structured) set of arrays, rather than changing real
;; memory.

(COMS ;; Page creation, locking/unlocking,

(FNS MKI.NEXTPAGE WriteoutPage MKI.NEWPAGE MKI.LOCKPAGES MKI.LOCKEDPAGEP RESETMEMORY))
(DECLARE%: EVAL@COMPILE (VARS (PTRNIL 0)
(PTRNOBIND 1))

DONTCOPY
(RECORDS ALTOPOINTER))
(VARS (NEWPAGETRACE)
(BLANKPAGE))

(COMS ;; Functions for fetching and replacing words, pointers, and FIXPs in the MAKEINIT. If, e.g., we need GETBASEFLOATP in the init,
;; we'd have to add a definition here. You'd also need to add an entry to MKI.SUBFNS (e.g. (\GETBASEFLOATP .
;; I.GETBASEFLOATP)), so that the renaming code would catch it.

(FNS I.PUTBASE I.GETBASE I.PUTBASEPTR I.GETBASEPTR I.GETBASEFIXP I.PUTBASEFIXP)
(DECLARE%: DONTCOPY (MACROS .DOADDBASE. I.HILOC I.LOLOC I.ADDBASE I.PAGELOC I.VAG2)))

;; MEMARRAY is the fake-memory array.

(VARS (MEMARRAY))
(GLOBALVARS MEMARRAY NONPAGE NONPAGE2 PTRNIL BLANKPAGE NEWPAGETRACE)))

;; Memory-faking functions for use in MAKEINIT. These read and write from a huge (tree-structured) set of arrays, rather than changing real memory.

;; Page creation, locking/unlocking,

(DEFINEQ

(**MKI.NEXTPAGE**

[LAMBDA (VP)

; Edited 5-Nov-92 15:40 by sybalsky:mv:envos

;; Find the next occupied virtual page, starting with VP. Can return VP, if it exists.

;; Used only in MAKEINIT.

(PROG (A I)
L1 (COND
((IGREATERP VP 65535)
(RETURN))
((EQ (SETQ A (FASTELT MEMARRAY (LRSH VP 8)))
NONPAGE2)
(SETQ VP (ADD1 (LOGOR VP 255)))
(GO L1)))
(SETQ I (LOGAND VP 255))
L2 [COND
((EQ I 256)
(SETQ VP (ADD1 (LOGOR VP 255)))
(GO L1))
((NEQ (FASTELT A I)
NONPAGE)
(RETURN (IPLUS (LOGAND VP 65280)
I])
(SETQ I (ADD1 I))
(GO L2])

(**WriteoutPage**

[LAMBDA (FX VP)

; Edited 26-Oct-92 15:23 by sybalsky:mv:envos
(* For MAXC)

;; Write out the page VP to file FX. Used only for MAXC, according to above comment (not verified)??

(PROG [(A (FASTELT (FASTELT MEMARRAY (LRSH VP 8))
(LOGAND VP 255)
(AOUT A 0 256 FX 'SMALLPOSP))

(**MKI.NEWPAGE**

[LAMBDA (PTR NOERROR LOCK? BLANKFLG)

; Edited 5-Nov-92 15:49 by sybalsky:mv:envos

;; Create a new virtual page for the MAKEINIT.

```

(PROG (A LO1 PAGE)
[COND
  ((EQ (SETQ A (FASTSET MEMARRAY (LRSH PTR 16)))
    NONPAGE2)
    ;; This whole segment hasn't been created yet. Create it as a segment full of non-page entries.
    (FASTSETA MEMARRAY (LRSH PTR 16)
      (SETQ A (POINTERARRAY 256 NONPAGE]
[COND
  ((NEQ (FASTSET A (SETQ LO1 (LRSH (LOGAND PTR 65535)
    8)))
    NONPAGE)
    ;; This page better not yet have been allocated!
    (HELP PTR "already allocated"]
(COND
  (NEWPAGETRACE (printout NEWPAGETRACE "page " .I3.8 (I.HILOC PTR)
    ", " .I3.8 (LRSH (I.LOLOC PTR)
    8)
    %,,)))
  ;; Fill in the MEMARRAY entry for this page with a 256-word block of nothing much. If BLANKPAGE, fill it in with the same page of blanks.
  [FASTSETA A LO1 (COND
    ((NOT BLANKFLG)
      (WORDARRAY 256))
    (T (OR BLANKPAGE (SETQ BLANKPAGE (WORDARRAY 256]
  ;; If LOCK?, this is a locked page, so add it to the locked page table.
  (AND LOCK? (MKI.LOCKPAGES PTR 1))
  (RETURN PTR])

```

(MKI.LOCKPAGES

```

[LAMBDA (PTR NPAGES) (* Imm "11-AUG-80 21:53")
  (push LOCKEDPAGES (CONS (I.PAGELOC PTR)
    NPAGES])

```

(MKI.LOCKEDPAGEP

```

[LAMBDA (VP) (* Imm "9-FEB-82 21:54")
  (for X in LOCKEDPAGES when [AND (IGEQ VP (CAR X))
    (ILESSP VP (IPLUS (CAR X)
      (CDR X))
    do (RETURN T])

```

(RESETMEMORY

```

[LAMBDA NIL (* Imm "26-MAR-81 09:23")
  (SETQ LOCKEDPAGES)
  (COND
    ((NULL MEMARRAY)
      (SETQ NONPAGE (WORDARRAY 256))
      (SETQ NONPAGE2 (POINTERARRAY 256 NONPAGE))
      (SETQ MEMARRAY (POINTERARRAY 256 NONPAGE2)))
    (T (for I from 0 to 255 do (FASTSETA MEMARRAY I NONPAGE2])
  )

```

```

(DECLARE%: EVAL@COMPILE

```

```

(RPAQQ PTRNIL 0)

```

```

(RPAQQ PTRNOBIND 1)

```

```

(DECLARE%: EVAL@COMPILE

```

```

[ACCESSFNS ALTOPOINTER ((HILOC (LRSH DATUM 16))
  (LOLOC (LOGAND DATUM 65535)))
  (CREATE (IPLUS (LLSH HILOC 16)
    LOLOC))
  (ACCESSFNS ALTOPOINTER ((6to13 (LRSH (LOGAND DATUM 65535)
    8))
    [bit12 (NOT (ZEROP (LOGAND 512 DATUM]
    (0to13 (I.PAGELOC DATUM))
    (0to11 (LRSH (I.PAGELOC DATUM)
    2]
  )
  )

```

```

(RPAQQ NEWPAGETRACE NIL)

```

```

(RPAQQ BLANKPAGE NIL)

```

```

;; Functions for fetching and replacing words, pointers, and FIXPs in the MAKEINIT. If, e.g., we need GETBASEFLOATP in the init, we'd have to add a
;; definition here. You'd also need to add an entry to MKI.SUBFNS (e.g. (\GETBASEFLOATP . I.GETBASEFLOATP) ), so that the renaming code would
;; catch it.

```

(DEFINEQ

(I.PUTBASE

```
[LAMBDA (PTR D V)
  (PROG (HI LO1 LO2)
    (DECLARE (LOCALVARS . T))
    (..DOADDBASE. PTR D)
    (AND (EQ (SETQ HI (FASTELT (FASTELT MEMARRAY HI)
                                LO1))
            NONPAGE)
          (INVALIDADDR))
    (RETURN (FASTSETAW HI LO2 V]))
```

(I.GETBASE

```
[LAMBDA (PTR D)
  (PROG (HI LO1 LO2)
    (DECLARE (LOCALVARS HI LO1 LO2))
    (..DOADDBASE. PTR D)
    (COND
      ((EQ (SETQ HI (FASTELT (FASTELT MEMARRAY HI)
                              LO1))
            NONPAGE)
        (INVALIDADDR)))
    (RETURN (FASTELOW HI LO2]))
```

(I.PUTBASEPTR

[LAMBDA (PTR D V)

; Edited 26-Oct-92 15:19 by sybalsky:mv:envos

;; MAKEINIT version of \PUTBASEPTR. Must be able to handle local symbols, so \CREATE.SYMBOL can set values to NOBIND. (JDS 10/26/92).

```
(PROG (HI LO1 LO2 (VAL (OR V PTRNIL)))
  (DECLARE (LOCALVARS . T))
  (..DOADDBASE. PTR D)
  (AND (EQ (SETQ HI (FASTELOW (FASTELOW MEMARRAY HI)
                              LO1))
            NONPAGE)
        (INVALIDADDR))
  (AND (LITATOM VAL)
        (SETQ VAL (I.ATOMNUMBER VAL)))
  (FASTSETAW HI LO2 (LRSH VAL 16))
  (FASTSETAW HI (ADD1 LO2)
    (LOGAND VAL 65535))
  (RETURN VAL]))
```

(I.GETBASEPTR

```
[LAMBDA (PTR D)
  (PROG (HI LO1 LO2)
    (DECLARE (LOCALVARS . T))
    (..DOADDBASE. PTR D)
    (COND
      ((EQ (SETQ HI (FASTELOW (FASTELOW MEMARRAY HI)
                              LO1))
            NONPAGE)
        (INVALIDADDR)))
    (RETURN (I.VAG2 (LOGAND 255 (FASTELOW HI LO2))
      (FASTELOW HI (ADD1 LO2]))
```

(I.GETBASEFIXP

[LAMBDA (PTR D)

; Edited 26-Oct-92 13:33 by sybalsky:mv:envos

;; MAKEINIT version of \GETBASEFIXP.

```
(PROG (HI LO1 LO2)
  (DECLARE (LOCALVARS HI LO1 LO2))
  (..DOADDBASE. PTR D)
  (COND
    ((EQ (SETQ HI (FASTELOW (FASTELOW MEMARRAY HI)
                            LO1))
          NONPAGE)
      (INVALIDADDR)))
  (RETURN (IPLUS (LLSH (FASTELOW HI LO2)
    16)
    (FASTELOW HI (ADD1 LO2]))
```

(I.PUTBASEFIXP

[LAMBDA (PTR D V)

; Edited 26-Oct-92 12:30 by sybalsky:mv:envos

```
(PROG (HI LO1 LO2)
  (DECLARE (LOCALVARS . T))
  (..DOADDBASE. PTR D)
  (AND (EQ (SETQ HI (FASTELOW (FASTELOW MEMARRAY HI)
                              LO1))
            NONPAGE)
        (INVALIDADDR))
```

```

    (RETURN (PROGN (FASTSETAW HI LO2 (LRSH V 16))
                    (FASTSETAW HI (ADD1 LO2)
                                   (LOGAND V 65535]))
)

(DECLARE%: DONTCOPY

(DECLARE%: EVAL@COMPILE

[PROGN (PUTPROPS .DOADDBASE. MACRO ((PTR D)
                                   (ASSEMBLE NIL (CQ (VAG (IPLUS PTR D)))
                                   (LSHC 1 %, -16)
                                   (ADDI 1 %, ASZ)
                                   (SETQ HI)
                                   (LSHC 1 %, 8)
                                   (ANDI 1 %, 255)
                                   (ADDI 1 %, ASZ)
                                   (SETQ LO1)
                                   (LSHC 1 %, 8)
                                   (ANDI 1 %, 255)
                                   (ADDI 1 %, ASZ)
                                   (SETQ LO2)))
      (PUTPROPS .DOADDBASE. ALTOMACRO ((PTR D)
                                       (SETQ HI (LRSH (SETQ LO2 (IPLUS PTR D))
                                                         16))
                                       (SETQ LO1 (LOGAND 255 (LRSH LO2 8)))
                                       (SETQ LO2 (LOGAND LO2 255))))
      (PUTPROPS .DOADDBASE. DMACRO ((PTR D)
                                    (SETQ HI (LRSH (SETQ LO2 (IPLUS PTR D))
                                                         16))
                                    (SETQ LO1 (LOGAND 255 (LRSH LO2 8)))
                                    (SETQ LO2 (LOGAND LO2 255))))]

(PUTPROPS I.HILOC MACRO ((PTR)
                          (LRSH (OR PTR PTRNIL)
                                16)))

(PUTPROPS I.LOLOC MACRO ((PTR)
                          (LOGAND (OR PTR PTRNIL)
                                65535)))

(PUTPROPS I.ADDBASE MACRO ((PTR D)
                            (IPLUS (OR PTR PTRNIL)
                                   D)))

(PUTPROPS I.PAGELOC MACRO ((PTR)
                            (LRSH (OR PTR PTRNIL)
                                   8)))

(PUTPROPS I.VAG2 MACRO ((HI LO)
                        ([LAMBDA (X)
                          (DECLARE (LOCALVARS . T))
                          (COND
                           ((ZEROP X)
                            NIL)
                           (T X]
                          (IPLUS (LLSH HI 16)
                                LO))))
)
)

;; MEMARRAY is the fake-memory array.

(RPAQQ MEMARRAY NIL)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS MEMARRAY NONPAGE NONPAGE2 PTRNIL BLANKPAGE NEWPAGETRACE)
)

(PUTPROPS MEM COPYRIGHT ("Venue & Xerox Corporation" 1982 1983 1986 1990 1992))

```

FUNCTION INDEX

I.GETBASE	3	I.PUTBASE	3	MKI.LOCKEDPAGEP	2	MKI.NEPTPAGE	1
I.GETBASEFIXP	3	I.PUTBASEFIXP	3	MKI.LOCKPAGES	2	RESETMEMORY	2
I.GETBASEPTR	3	I.PUTBASEPTR	3	MKI.NEWPAGE	1	WriteoutPage	1

VARIABLE INDEX

BLANKPAGE	2	MEMARRAY	4	NEWPAGETRACE	2	PTRNIL	2	PTRNOBIND	2
-----------------	---	----------------	---	--------------------	---	--------------	---	-----------------	---

MACRO INDEX

I.ADDBASE	4	I.HILOLOC	4	I.LOLOLOC	4	I.PAGELOC	4	I.VAG2	4
-----------------	---	-----------------	---	-----------------	---	-----------------	---	--------------	---

RECORD INDEX

ALTOPOINTER	2
-------------------	---
