Lisp and Mac OS X

October 2003, ILC 2003 Rainer Joswig, joswig@lisp.de

- Mac OS X is a very popular desktop OS
- Mac OS X and Xserve are very cost/space effective servers
- Powerbook G4 is my personal Lisp workstation of choice
- PowerMac G4 (G5) with Cinema Display makes developing Lisp code fun
- any eMac/iMac/iBook is enough for Lisp development and Lisp applications
- a multitude of Lisp systems is available for Mac OS X

Mac OS X

- Mac OS 9 is history. Rich environment on an unstable OS. Great UI. I don't have it anymore installed.
- Mac OS X is based on Mach, Unix, Carbon, Cocoa, Java, FreeBSD, OpenGL. Stable. Fast. Great UI.
- Darwin is an open source version of the core Mac OS X. Runs on PowerPC and x86.
- Server version of Mac OS X is available
- compatible to a wide variety of standards.

Mac OS X 10.2 "Jaguar"

- Quartz Extreme (OpenGL accelerated compositing of Windows)
- many speed improvements
- first really "usable" version of Mac OS X

Mac OS X 10.3 "Panther"

- lot's of UI improvements
- GCC 3.3
- XCODE IDE
- new Finder
- early support for PowerMac G5
- appears 24th October 2003



▲ 40 Thu 7:06 PM

d Xcode File Edit Format View Find Project Build Debug Data SCM Window Help

Mac OS X



Mac OS X system architecture

Carbon

- Port of the Mac OS 9 libraries to Mac OS X. Simplifies porting of Mac OS 9 software to Mac OS X.
- some functionality has been removed or changed
- some extensions
- Macintosh Common Lisp is a Carbon application

Cocoa

- the new Mac OS X libraries
- based on Objective C
- GC: Reference Counting
- Message Passing
- Cocoa-based applications can use Carbon
- OpenMCL and LispWorks can use Cocoa

new in Mac OS X

- Unix (FreeBSD 4.4) with Mach Kernel
- command line interface, shell
- multiple users (with live switching)
- long pathnames
- sockets
- OpenGL/Quartz/Aqua User Interface
- memory protection
- preemptive multi-tasking, threads
- multi-processor support
- Apache, sendmail, GCC 3.3, ...
- journalling filesystem

still in Mac OS X

- Unicode
- AppleEvents
- OpenGL
- AppleScript
- HFS+

PowerPC generations

- G3, power efficient, great for Laptops, runs Lisp very nicely
- G4, new versions are also usable for Laptops, runs Lisp very nicely, AltiVec extension (only some support by Lisp implementations)
- G5, not yet usable for Laptops, 64bit processor, currently many Lisp implementations seem to have problems (code runs slow)

Common Lisp

- Lisp started in 1958, introduced many concepts (many still to enter mainstream)
- Evolution: ANSI Common Lisp
- based on decades of experience
- some very large applications (> 1 million LOCs) and some extremely large applications (like Cyc)
- advanced Object System (CLOS) -> OODL (Object-Oriented Dynamic Language)

extremely powerful Object-oriented dynamic language (OODL)

- CLOS dynamic objects allow changes at any time (read, compile, load, run)
- first class metaobjects (class, function, method, metaclass, ...)
- add/delete/redefine class, objects can change their classes
- change inheritance of classes
- classes are instances of metaclasses
- add/remove/redefine slots
- add/delete/redefine methods and functions
- method combination, multi-methods, multiple-inheritance
- methods for objects
- introspection, meta object protocol (MOP)
- compiler and debugger available at runtime, load code at runtime

Common Lisp history on the Mac

- available on the Mac for many years
 - Pearl Lisp, Coral Lisp, Coral Common Lisp, Macintosh Allegro Common Lisp, Macintosh Common Lisp, OpenMCL
 - once owned by Apple for a few years
 - used for a lot of applications (Apple Dylan, Sk8, OpenMusic, Symbolic Composer, Igor Engraver, ...) and teaching CS
 - PowerLisp (obsolete)
 - Procyon Common Lisp (obsolete)
 - Exper Common Lisp (obsolete)

http://openmcl.clozure.com/ http://www.digitool.com/ http://www.cormanlisp.com/PowerLisp.html

MCL 5.0

- from Digitool, http://www.digitool.com/
- "carbonized" port of the popular Macintosh Common Lisp
- final version since mid 2003
- compiles to native PowerPC code and can generate applications
- Integrated Development Environment with Interface Builder and Editor (FRED)
- supports cooperative threads, OpenTransport, AppleEvents, ...
- CLIM 2.0 as option

MCL 5.0 (2)

- lot's of libraries (Quicktime, OpenGL, ...)
- still also runs under Mac OS 9
- still: old event loop, short pathnames, no Cocoa support, ...
- http://www.digitool.com/
- ftp://ftp.digitool.com/

MCL 5.0

discription of the second state of the second	🗊 V	৵ — B: 670MB 0 ↓↓↓↓ 10	0% 💻 🛜 🌒 📼 🕅	99%) Fr., 13:14:52 Uhr
OOO plot.lisp {PopLife:Lisp:MCL:MCL 5.0:CLIM:Demo:}	000	Pl	ot Demo	
;;; CLIM based plotting package	Plot			Tokyo
<pre>(defmacro plotting-data ((stream %rest options) &body body) ;; For each X value we want to specify the Y values ;; Specify label for each X value ;; Specify label for each set of Y values ;; Specify label for each set of Y values ;; Specify label for each set of Y values ;; Specify label for each set of Y values ;; Pange of values for the axiss (let ((plott-ing-continuation (gensym)))</pre>	20.			New York
(define-presentation-type graph-plot ())	1			
<pre>(define=presentation=type graph=region ()) (define=presentation=method accept ((type graph=region) stream (view textual=view) &key)</pre>	<i>Options</i> Graph type: Min X: Min Y: Max X: Max Y:	• Plot Bar Pie None None None None		
<pre>(define-presentation-method present (region (vipe graph-region) stream (view textual-view) areg) (with-bounding-rectangle* (left top right bottom)) (present `((,left,top) (,right ,bottom)) ((sequence-enumerated real) :echo-space nil) (sequence-enumerated real) :echo-space nil)) :separator #: echo-space nil) :stream stream))) (define-presentation-method presentation-typep (object (type graph-region)) (rectangle object))</pre>	20069 60 1960 70 1970 80 980 type:	Mexico City Tokyo 5 11 8 15 14 Apropos ³ FUNCTION	New York 14 16 15.5	Demos <i>CLIM Demonstrations</i> 15 Puzzle Address Book Bicycle Gearing
	package:	CLIM-DEMO	apropos	Bitmap Editor
O O O Listener ? (apropos "start-demo") UN-DEND START-DEND, Deft. FUNCTION	contains:	demo	inspect	CAD Demo Color Chooser Cutom Output Percente
? (clis-demoistart-demo) CL-USER Busy	DEFINE-CAD-C DEFINE-CAD-C DEFINE-CAD-C DEFINE-COEMO DEFINE-COEMO DEFINE-PLOT- DEMO-SLEEP DO-GRAPHICS- DO-GRAPHICS- DO-PLOT-DEMC	EHO-COMMAND II CS-DEMO-COMMAND DEMO-COMMAND DEMO Symbols only	callers doc methods initargs	Flight Planner Graphics Demos Lisp Listener Peek Plotting Demo Thinkadot Exit

Allegro Common Lisp 6.2

- from Franz Inc., http://www.franz.com/
- port of the widely available Unix version
- compiles to native PowerPC code
- Does not have a Mac GUI or IDE
 - IDE is Emacs + ELI + Composer (X11)
- evaluation version for Mac OS X is available

Allegro Common Lisp 6.2

Emacs File Edit Options Buffers Tools ACLFile ACLEdit ACLDe	bug ACLHelp Composer Help 🕴 S74M8 📖 27% 🧮 奈 🜒 💽 (99%) Sa., 23:33:47 Uhr
🖯 🖯 🖯 Emacs@RJPBG4.local.	
;; ':EUC-BASE'	X Process Browser
CL-USER(2): (defclass foo () (a b))	Window
#-STANDARD-CLASS FOD> CL-USER(3): (defclass bar (foo) (c))	* 12 0 23 0.5 100 inactive process browser event handler,
#GTANDARD-CLASS BARS CL_USER(4): (defalats haz (foo) (d))	* 11 2 5 0.0 0 runable Process Browser
#-STANDARD-CLASS BAZ>	* 6 9 10 0.0 536870910 waiting x11 event dispatcher, waiting for input
CL-USER(5): (defun test-it () (break))	 1 0 14 0.0 0 waiting Initial Lisp Listener, waiting for input 3 0 0 0.0 0 waiting Connect to Enacs daemon, waiting for input
CL-USER(6): (compile *)	* 5 0 4 0.0 0 waiting Editor Server, waiting for input 7 4 0 0 0.0 0 waiting Editor Server, waiting for input
TEST-IT	* 10 0 0 0.0 0 waiting Domain Name Server Client, waiting for input
NIL	* 7 0 0 0.0 100 inactive Default Window Stream Event Handler, arrested for event mone control
CL-USER(7); (test-it) Break: call to the break' function	* 8 0 10 0.0 100 inactive Grapher Event Handler,
brown out to the brown function.	* 14 0 10 0 0 0 maiting framework waiting for Pauhard mont
Restart actions (select using :continue):	
1: Return to Top Level (an "abort" restart).	A A A A A A A A A A A A A A A A A A A
2: Abort entirely from this process.	X CLOS Class Hierarchy for FOO
[1C] LL-USER(9): CL-USER(9): (test-it)	Window History
Break: call to the 'break' function.	
Restart actions (select using scontinue):	FOO
0: return from break.	BAR
1: Return to Top Level (an "abort" restart).	
[1c] CL-USER(10): :continue 1	
CL-USER(11): (test-it) Break: call to the 'break' function.	Para validati na nanana na nanana Referen Planung FHO interation Plan
	X Inspector for: # <standard-llass bar=""></standard-llass>
Restart actions (select using continue): 8: return from break.	Window Inspect as History
1: Return to Top Level (an "abort" restart).	Instrumentary way in the oldes over the original second second second second second second second second second
2: Abort entirely from this process. fic1 CL-USER(12): :wdebwa	The class is not get finalized.
	Its direct superclass is F00. It has no subclasses.
Bringing up a vindov debugger for #dulTIPROCESSING:PROCESS Initial Lisp Listemer[1] 0 #3816cc2a>	Lo. Its 1 instance slot is
Window debugger is now active.	6 defined in class BAR.
(1c) CL-USER(13):	
-1:**-Emacs ALL Idle *common-lisp* Bot L91 (Inferior Common Lisp) #-STANDARD-CLASS BAZ> is the class named BAZ of the metaclass STANDARD-CLASS.	COO Debugges for total tracks Lessons Le akolida
The class is not yet finalized.	X Debugger for: Initial Lisp Listener
Its alreat superclass is FOO. It has no subclasses.	Window Debugger
Its 1 instance slot is	(IREAK)
D defined in class BAZ.[]	-> (TEST-IT)
	(IVAL (ISJT-IT)) (TOP-LEVEL::READ-EVAL-PRINT-OME-COMMAND NIL NIL)
	(EXCL:: BEAD-EVAL-PRINT-LOOP :LEVEL 0) (TAD-EVAL-TAD-EVAL-PRINT-LOOP :LEVEL 0)
	(TOP-LEVEL. TOP-LEVEL-BEAD-EVAL-PRINT-LOOP)
	(SYSTEM::RUNTIME-OPERATION #(Punction TOP-LEVEL-PEAD-EVAL-PPINT-LOOP) NIL) (TOP-LEVEL:START-INTERACTIVE-TOP-LEVEL
-1:**-Emacs ACL Idle *CL-temp* All L6 (Common Lisp)	A TERMINAL COMPANY CONTRACT CONTRACT OF A ALL A ANDALISA
Quit Centres Concept Verters	

OpenMCL 0.13.6 / 0.14 alpha

- from Clozure, based on MCL from Digitool, http:// /www.clozure.com/
- initially developed for NASA by Gary Byers for server and embedded use (-> small footprint)
- solid **free** and open source core of Macintosh Common Lisp
- currently under active development by Gary Byers
- also runs on PPC Linux, older ports for SPARC/ Solaris and QNX
- compiles to native PowerPC code
- usually used with Emacs/ilisp

OpenMCL 0.13.6 / 0.14 alpha

- supports threads, sockets (ACL API), CLX, ...
- (simple) Cocoa-IDE is under development
- can be used for double-clickable applications under Mac OS X (example Alpaca)
- OpenMCL 0.14 alpha
 - native threads with support for multiple processors
 - MOP
- a CLX version also runs McCLIM
- work in progress
- EGC does not work yet

OpenMCL



Mikel Evin's Alpaca

🧉 Alpaca File Edit Format Lisp	Window Help	🗐 📢 🚸 — 🖁 676MB 힟 业 18% 💻	🛜 🜒 📧 (99%) Fr., 12:32:14 Uhr
🖲 🖯 🔵 New RTF Document 🛛 🕷	SN tled	Alpaca	
New Text Document 企業	IN		
Open #	80		11 💚 🛧 »
Open Recent	>	Zurück Vor Darstellung Pfad Computer P	rivat Favoriten Programme
Close	20/	23 Objekte, 919,4 MB v	erfügbar
Save #	25	Name	▲ Änderungsdatum
		🔻 🧊 Contents	Heute, 12:23 Uhr
Bovert		CVS	Heute, 12:23 Uhr
Reven		Info.plist	Heute, 12:23 Uhr
Page Setup 🏠	P	🔻 🐺 🎾 MacOS	Heute, 12:23 Uhr
Print #	3P	🗋 Alpaca	Heute, 12:23 Uhr
		📄 Alpaca.image	Heute, 12:23 Uhr
Network Working Group	P. Leach	CVS	Heute, 12:23 Uhr
Request for Comments: 2831 Category: Standards Track	C Newman	PkgInfo	Heute, 12:23 Uhr
Innos	oft	🔻 🧊 Resources	Heute, 12:23 Uhr
May 2	2000	👰 Alpaca.icns	Heute, 12:23 Uhr
		CVS	Heute, 12:23 Uhr
Lising Digest Authentication as a S	SASI Mechanism	🔻 🧊 English.lproj	Heute, 12:23 Uhr
osing Digost Automodulon as a c		AlpacaEditor.nib	Heute, 12:23 Uhr
Status of this Memo		AlpacaRTFEditor.nib	Heute, 12:23 Uhr
This desument enseifies an internet ste	ndarda traak protocol for the	AlpacaTextEditor.nib	Heute, 12:23 Uhr
Internet community, and requests discu	ssion and suggestions for	Credits.rtf	Heute, 12:23 Uhr
improvements. Please refer to the curre	ent edition of the "Internet	🕨 🕨 🧊 CVS	Heute, 12:23 Uhr
Official Protocol Standards" (STD 1) for	the standardization state	InfoPlist.strings	Heute, 12:23 Uhr
and status of this protocol. Distribution	of this memo is unlimited.	lispeditor.nib	Heute, 12:23 Uhr
Copyright Notice		🚼 MainMenu.nib	Heute, 12:23 Uhr
		Copenmclinspector.nib	Heute, 12:23 Uhr
Copyright (C) The Internet Society (200	0). All Rights Reserved.	🕨 🧊 Help	Heute, 12:23 Uhr
Abstract		🕨 🕨 💭 CVS	Heute, 12:23 Uhr
Abstract			
This specification defines how HTTP Di	gest Authentication [Digest]		
can be used as a SASL [RFC 2222] me	chanism for any protocol that has		
[RFC 2195] and as a convenient way to	support a single authentication	Control Control 1.000000 seconds Listener-2	UK_0212doc
mechanism for web, mail, LDAP, and of	ther protocols.	Welcome to OpenMCL Version (Beta: Darwin) 0.13!	
Table of Contents			
Table of Contents			
	1		
	1		
-			
		CL-USER	

- from Xanalys, http://www.lispworks.com/
- very feature-complete Lisp implementation, also available for Unix/ Linux and Windows
- compiles to native code
- LispWorks for Mac OS X Personal Edition available now
- comes in three flavors:
 - command line
 - CAPI via Cocoa
 - X11/Motif version, also supports CLIM 2.0
- Editor provided in source
- Extensive IDE

- lacks support for Mac libraries (Quicktime, MIDI, CoreAudio AppleEvents, ...)
- no CLIM for Cocoa-LispWorks
- CAPI is native on the Mac **and** cross-platform (Unix, Linux, Windows, Mac)

莺 LispWorks1 File Edit Expression Values Debug History Window Help	🗐 😵 🚸 — 🗄 659MB 🕛 💷 38% 🥅 🛜 📢 💽 (96%) Fr., 15:14:51 Uhr
🐚 🖻 🔍 🥰 🛱 🖹 🕊 🚳 🕫 🕫 🐼 🖃 🗮 🚬 🔍 📁 📕	COO Editor 15 - capi-prompt-for-password.lisp
CL-HTTP 70.173 [LW 1:8.6] (LispWorks 4.3.6)	
Start Stop Inspect Start Stop Start	Text Output Buffers Definitions Changed Defi Find Definitio
Expose Log Window Edit Log File Edit Configuration	(:panes (user-name-pane capi:text-input-pane) (password-pane capi:password-pane))
Debug Server 🗹 Debug Client 📃 Debug Mailer	(:layouts (main-layout capi:grid-layout
Resolve IP Addresses 🗹 Resolve URL Hostnames 🗌 Resolve IP Addresses for Logging	"Password:" password-pane :columns 2)
Edit CL-HTTP File Load System Compile System Compile and Load System hpProoffindPrint	(:default-initargs :layout 'main-layout))
O O O Listener 3	#+CAPI (defmethod initialize-instance :after ((self gui-ask-password-dialog)
🗅 💕 📮 🐰 🖻 🛍 🍋 🎦 🔮 Enter Username and Password:	(with-slots (user-name-pane password-pane) self (when name
User Name: joswig	(setf (capi:text-input-pane-text user-name-pane) name)) (when password
CL-USER 13 : 1 > :top Password:	(setf (capi:text-input-pane-text password-pane) password))))
CL-USER 14 > nil NIL Cancel	(defun gui-ask-password-dialog-name (self) (with-slots (user-name-pane) self
CL-USER 15 > nil	(capi:text-input-pane-text user-name-pane)))
NIL	#+CAP1 (defun gui-ask-password-dialog-password (self) (with-slots (nassword-name) self
e" "joma@ai.mit.edu" "CL-HTTP runs!" "" :user user :password password)) П	(capi:text-input-pane-text password-pane)))
	#+CAPI (defun capi::prompt-for-password (&optional name pwd)
	(let ((dialog (make-instance 'gut-ask-password-atalog :title "Enter Username and Password:"
	:password pwd))) (if (capi:popup-confirmer dialog nil)
	(values (gui-ask-password-dialog-name dialog) (gui-ask-password-dialog-password dialog)
Oc 19:23 Jost HindowServer [174] And wed range exhausted. (0xbbdb3000 to 0xbbd19:000 00 19:51 Advist HindowServer [174] And wed range exhausted. (0.0 HTTP Console Log	(values nil t))))
h-0; en-US; rv:1.5) Gecko/20030916" -	; (capi::prompt-for-password nil nil)
<pre>[2003-10-10 12:30:36] {0.01 0} 127.0.0.1 304 HTTP/1.1 8000 {1} 251 - "GET /cl-http/icons/power.gif" "Mozilla/5.0 /cl-http/projects.html"</pre>	;;;; EOF
2003-10-10 12:30:36] {0.0 0} 127.0.0.1 304 HTTP/1.1 8000 {2} 251 - "GET /cl-http/icons/line-rain.gif" "Mozilla/ 000/cl-http/projects.html" П	Y I
	LATIN-1 capi-prompt-for-password.lisp {CL-USER} (Lisp) 13+87 /Lisp/LispWorks/capi-
Ready.	

LispWorks File Edit View Buffers Definitions Expression History Win	dow Help 💶 24% 💻 🛤 🔿 🛒 🖘 📢 💽 (100%) Fr., 23:29:13 Rainer Joswig
Reset Shading Texture View Material Image: Shading Texture	
Object Light Ambient Image: Image	Editor 4 - icosahedron.lisp
Mouse-L spins the object, Mouse-R moves the light, Shift Mouse-L moves your view.	(defin gl-vertexes (contents) (napcar #'(lambda (c) (apply 'gl-double-vector c)) contents))

di X11 Applications Edit Window Help	🗐 👽 🚸 🖡 👬 🔜 100% 🥅 🛜 🖌 💽 (32%) Fr., 2:16:27 Uhr
000 x LispWorks 4.3.0 on RJPBG4.local.	Browser
Break to tty Works File Tools Windows Help	Browser Display Snapshot
Interrupt Lisp Quit Golonitor Idle	CUM SUCA GENERIC SCROLLER PANE SING CUM SUCA GENERIC SCROLLER PANE SING UM SUCA GENERIC SCROLLER PANE SUCA GENERIC SCROLLER PANE CUM SUCA GENERIC SCROLLER PANE CUM SUCA GENERIC SCROLLER PANE CUM SUCA GENERIC SCROLLER PANE CUM SUCA GENERIC SCROLLER PANE
Kustener 1 Works File Edit Expression Values Debug History Help Image:	CUM SUCA SCHARMEN CUM SUCA COMPOSITE PANE CUM SUCA SCHOLE SAR PANE C
Ro open bracket	Comawand: Show Graph CLIM: PANE Comawand:] Starting depth: 3
	Enter an integer between 10 and 20:
CL-USER 1 >	IL OK Cancel (3 11) ⇒ (LET (A B) (ACCEPTINO-VALUES (*QUERY-IO*) (SETQ A (ACCEPT 'INTEGER)) (TERP RI *QUERY-IO*) (SETQ B (ACCEPT '(INTEGER 10 20)))) (LIST A B)) Enter an integer: NIL Enter an integer between 10 and 20: NIL OK Cancel
Ready. QUSER 1 > []	

CLISP 2.31

- portable open source Common Lisp under active development
- written mostly in C
- single-threaded, byte-code compiler
- can be used for scripting
- gettting a working/compilable version can be difficult
- http://clisp.cons.org/

ThinLisp

- originally from Gensym, now Open Source
- small CL subset used for delivery and realtime programming
- compiles to C
- uses other Lisps as a host
- http://www.thinlisp.org/
- no closures, no GC

Embeddable Common Lisp (ECLS)

- Current version 0.8
- compiles to C
- ported to Mac OS X, has some rough edges
- http://ecls.sourceforge.net/
- embeddable

Embeddable Common Lisp (ECLS)



ECLS embedded in Quake

Steel Bank Common Lisp (SBCL)

- Current version 0.8.4.x
- forked from CMUCL, simplified build
- can be compiled with some other Common Lisp implementations like OpenMCL
- sophisticated native code compiler
- type inference and type checking
- no threads on generational GC in the current port, but in the works
- http://sbcl.sourceforge.net/

• GCL, GNU Common Lisp

- not yet fully ported, but people are working on it.
- http://www.gnu.org/software/gcl/gcl.html
- Poplog ?
- old Xlisp, originally from David Betz
 - XLispStat, Vista

Feature comparison

	ACL	LispWorks	MCL	CLISP	OpenMCL	SBCL
	commercial	commercial	commercial	modified GPL	free	free
IDE	Emacs, ELI, Composer	Native	Native	Emacs, ILisp	Emacs, ILisp	Emacs, ILisp
UI support	X11	X11, Cocoa	Carbon	X11	X11, Cocoa	X11
use in shell	yes	yes	no	yes	yes	yes
compiler	native	native	native	byte-code	native	native
threads	Lisp	Lisp	cooperative	no	native	no
GC	generational	generational	EGC, generational	?	generational	mark & sweep
delivery	?	images, applications	images, applications	?	images, applications	?
CLIM	yes	yes (X11)	yes		McCLIM	
Price	\$?	\$999 PE, \$2999 EE	\$750			

Benchmarks

- Benchmarks by Eric Marsden. http://www.chez.com/ emarsden/downloads/cl-bench.tar.gz
- ran on a PowerBook G4, 800Mhz, 1 GB RAM
- no surprise: native code compilers are the fastest
- overall fastest is LispWorks, followed by ACL and SBCL
- ACL has only a few problems with some benchmarks
- SBCL needs some speed improvements for CLOS
- CLISP has very fast bignum routines
- OpenMCL's compiler generates compact code, but isn't very sophisticated otherwise

Benchmarks

Benchmark	LispWorks 4.3	Allegro CL 6.2	SBCL 0.8.3.38	OpenMCL 0.14a	CLISP 2.31
BOYER	5.13	3.86	1.69	3.75	21.31
BROWSE	0.72	0.77	0.86	1.41	4.25
DDERVIV	0.9	0.96	0.89	1.76	7.19
DERIV	1.57	1.64	1.64	3.14	12.98
DESTRUCTIVE	0.78	1.1	1.27	1.53	10.95
DIV2-TEST-1	1.52	0.72	1.01	2.35	10.43
DIV2-TEST-2	0.77	0.36	0.53	1.2	5.23
FFT	0.18	0.12	0.17	7.73	16.33
FRPOLY/FIXNUM	1.12	1.04	0.77	1.25	9.96
FRPOLY/BIGNUM	2.61	1.86	1.68	2.82	8.69
FRPOLY/FLOAT	1.15	1.08	0.78	1.79	6.68
PUZZLE	0.6	5.73	0.33	5.71	17.11
CTAK	2.12	2.1	1.06	2.17	17.88
TAK	1.23	0.94	0.78	1.54	22.93
RTAK	1.22	0.92	0.78	1.52	22.93
TAKL	1.33	1.58	1.2	4.52	31.58
STAK	1.54	10.98	1.25	3.95	17.29

Benchmarks (II)

Benchmark	LispWorks 4.3	Allegro CL 6.2	SBCL 0.8.3.38	OpenMCL 0.14a	CLISP 2.31
FPRINT	0.71	1.88	4.91	4.35	6.63
TRAVERSE	4.71	2.86	4.41	3.92	55.23
TRIANGLE	2.38	4.33	2.07	3.55	81.87
RICHARDS	1.41	3.59	1.04	1.46	24.11
FACTORIAL	0.7	0.74	0.48	1.46	2.3
FIB	0.18	0.16	0.35	0.23	5.55
BIGNUM/ELEM-100-1000	1.28	1.23	1.33	2.32	0.17
BIGNUM/ELEM-1000-100	5.58	4.66	5.0	10.6	0.51
BIGNUM/ELEM-10000-1	10.36	11.58	5.14	16.34	0.47
BIGNUM/PARI-100-10	0.2	0.11	1.52	2.71	0.1
BIGNUM/PARI-200-5	0.95	0.42	25.19	33.24	0.4
HASH-STRINGS	0.08	2.32	0.09	0.24	0.52
HASH-INTEGERS	0.54	2.9	0.76	5.9	1.67
BOEHM-GC	6.77	18.12	8.08	13.1	75.45

Benchmarks (III)



Emacs, Xemacs

- various versions are available
- Emacs Lisp, single-threaded, byte-code compiler
- Mac OS X 10.2.x comes with GNU Emacs 21.1.1 (terminal version)
- very good Carbon/Cocoa-versions are have appeared, http://mindlube.com/products/emacs/index.html

 even GNUS works
- GNU Emacs 21.4 will support Mac OS X directly
- currently XEmacs only works with X11 on Mac OS X

ilisp

- free Interface for "inferior" Lisps for Emacs
- remote Lisp runs in a listener buffer
- Source code buffers with many features
 - highlighting
 - indenting
 - M-., arglist, evaluation, compilation, ...
- problem supporting multiple-threaded Lisps
- http://sourceforge.net/projects/ilisp/

Glossary

- GUI: graphical user interface
- IDE: Integrated development environment
- LOC: Lines of Code
- Lisp: Lisp is simply perfect
- CLOS: Common Lisp Object System
- CLIM: Common Lisp Interface Manager
- CAPI: Windowing Toolkit for LispWorks
- OODL: Object-oriented dynamic language
- MOP: Meta-Object Protocol