COS 126 Lecture 20: Applications

Operating systems (mass market, today)

- set of applications programs
    - mail
    - editor
    - database
    - spreadsheet
    - desktop publishing
    - internet browsing/publishing
    - ....

Drawback: applications are for

- corporate America
- home computing

Examples of basic tools for scientists/engineers

- emacs
- netscape/AltaVista (HTML)
- Maple/Mathematica
- awk/perl
- TeX/LaTeX
- PostScript
- C/C++/Java/Fortran

COS 126 skills give you "license" to use these

- converse not true!
- [beware of unlicensed users]

Use the mass-market tools, too

- [when they're better!]

emacs

- **Emacs Makes A Computer Slow**
- **Escape Meta Alt Control Shift**
- **Emacs Makers Are Crazy Sickos**
- **Emacs Makes All Computing Simple**
- **Emacs Makefiles Annihilate C-Shells**
- **Emacs May Allow Customized Screwups**
- **Emacs Manuals Are Cryptic and Surreal**
- **Eventually Munches All Computer Storage**
- **Eight Megabytes And Constantly Swapping**
- **Elsewhere Maybe All Commands are Simple**
- **Excellent Manuals Are Clearly Suppressed**
- **Emacs May Alienate Clients and Supporters**
- **Extended Macros Are Considered Superfluous**
- **Except by Middle Aged Computer Scientists**
- **Every Mode Accelerates Creation of Software**

Features

    full set of basic editing commands
    efficient keystroke codes
    extensible/customizable

Bottom line: efficient person-machine interaction

    try out various capabilities
    buy a manual
    surf the web

www.cgd.ucar.edu/gds/thibaud/Emacs/slides.html

HTML

Make yourself a home page:

mkdir public_html

chmod 0755 public_html

cd public_html

emacs index.html

put the following in index.html

```
<HTML>
<HEAD><TITLE>Your Name</TITLE></HEAD>
<BODY>
<H2>Your Name</H2>
<p>Anything else you'd like to say</p>
</BODY>
</HTML>
```

chmod 0755 index.html

in Netscape, open

www.princeton.~username

Voila!

Next steps:

add stuff (next slide); debug with "Reload"

copy and edit someone else's (it's public!)

surf the web (not worth buying manual)

step-by-step introduction

http://www.CS.Princeton.EDU/courses/archive/fall97/cs111/labs/introHTML

online manual

http://werbach.com/barebones/barebone_html.html

HyperText Markup Language

MARKUP:

metacommands specifying document structure

fonts

bold <i>italic</i> <tt>typewriter</tt>

headers

<h3>hypertext markup language</h3>

paragraph

<p>

It was the best of times; it was the worst of times;

</p>

itemized list

<ul>

** headers**

** paragraph**

** itemized list**

**
**

HYPERTEXT: text with hot links

<A HREF="URL"></A>

tables, graphics

Defacto standard: 1950s technology

ISSUES

* who writes HTML code? [person or program?]
* who decides appearance? [author or browser?]
* can we have a different standard? [too late?]
* content vs. form

Mathematical Applications

"Let the computer do it!"

- numerical computation
- symbolic manipulation
- graphs/plots

Many systems available

- tailored to application or general purpose?
- worth using new environment
  - for problem at hand?
- interfaces to systems in use
- quality of user interface
- extent of mathematical library
- compute the right answer?
- available on your machine?

C, FORTRAN, Java (with old code and libraries)
bc, dc
maple, mathematica, matlab

Wolfram, "Mathematica, A system
for doing math by computer"
Char, Geddes, Leong, Gonnet, et. al.
"Maple V Reference Manuals"
"First Leaves: Tutorial Intro to Maple V"

Simple examples (Maple)

numerical

```
> 2+2;
                                4
> p := x -> x^3+x^2-3*x +2;
                                 3    2
                  p := x -> x  + x  - 3 x + 2
> p(2.3333);
                             13.14754816
```

symbolic

```
> p(y^2+3);
                    2     3     2     2      2
                  (y  + 3)  + (y  + 3)  - 3 y  - 7
> simplify(");
                       6       4       2
                      y  + 10 y  + 30 y  + 29
```

typical sequence

```
> int(p(x)*cos(x),x);
.     3                     2
.    x  sin(x) + 3 cos(x) x  - 9 cos(x)
.                         2
.        - 9 x sin(x) + x  sin(x) + 2 cos(x) x
> subs(x=2,");
                      - 6 sin(2) + 7 cos(2)
> evalf(");
                           -8.368812417
> quit;
```

Numeric computation (Maple)

evaluate functions

```
> evalf(cot(.6 + Pi/8));
                              .6524523271
```

find roots

```
> fsolve(cot(x+Pi/8) = x, x);
                              .6218013732
```

number theory

```
> ifactor(1010101010101010101010);
       (2) (5) (11) (23) (513239) (21649) (8779) (4093)
> 2*5*11*23*513239*21649*8779*4093;
       1010101010101010101010
```

linear algebra

```
> A := array([[1,3,-4], [1,1,-2], [-1,-2,5]]);
                               [  1   3  -4 ]
                               [            ]
                          A := [  1   1  -2 ]
                               [            ]
                               [ -1  -2   5 ]
> Ainv := evalm(A^(-1));
                              [ -1/4   7/4  1/2 ]
                              [                 ]
                      Ainv := [  3/4  -1/4  1/2 ]
                              [                 ]
                              [  1/4   1/4  1/2 ]
```

Jacobian, Laplacian, etc.

Symbolic computation (Maple)

root-finding

```
> solve( x^4 - 7*x^3 + 3*a*x^2 = 0, x );
.                          1/2                    1/2
.  0, 0, 7/2+1/2 (49-12 a)   , 7/2-1/2 (49-12 a)
```

differential equations

```
> diff(y(x),x,x) + 2*diff(y(x),x) + y(x) = exp(-x);
.        2
.       d            d
.     ---- y(x) + 2 -- y(x) + y(x) = exp(- x)
.        2          dx
.      dx
> dsolve({", y(0)=1, D(y)(0)=0}, y(x));
.               2
.  y(x) = 1/2 x  exp(- x) + exp(- x) + exp(- x) x
```

series manipulations

```
> s := series(sin(x), x=0);
                            3          5      6
              s := x - 1/6 x  + 1/120 x  + O(x )
> c := series(cos(x), x=0);
                            2         4      6
              c := 1 - 1/2 x  + 1/24 x  + O(x )
> t := series(s/c, x=0);
                            3         5      6
              t := x + 1/3 x  + 2/15 x  + O(x )
```

Printing and typesetting

Typewriter

Keypunch + card reader + impact line printer

Traditional

- hot lead, galley proofs, etc.

WYSIWYG

- bitmap display editor
- laser printer
- ex: Bravo (1977) MacWrite, Word

"Batch"

- edited typescript
- typesetting software
- phototypesetter
- ex: vi + troff (1970x)
  - [emacs, etc.] + TeX (1979)

Internet publishing, hypertext

ALL INVOLVE: keyboard for input

Special challenges

- mathematical typesetting (TeX)
- hypertext (HTML)
- integrated fonts, graphics (PostScript)
- graphic art (Illustrator, Photoshop)
- standard representation (.pdf files)

Paperless, someday?

TeX

Ref: The TeXbook, Knuth.

INPUT: markup language

control sequences	\xxx
grouping	{ }
macros	\def
math formulae	$$
flexible space	\hskip\vskip
font control	\rm\it\bf
tables, alignments	\halign
page description	\eject

"programming" can be incidental

(but general functionality available)

OUTPUT:

long list of (x, y, character) records

advantages

- faithful to "art" of typsetting mathematics
- box-and-glue metaphor the "right" one

criticisms

- bitmap, nonstandard fonts
- no graphics
- baroque markup language (even LaTex)
- debugging, interaction weak

No reasonable alternative

for mathematical/technical publishing

Boxes and Glue

"Stuff" in TeX is in boxes, connected by glue

Horizontal, vertical, recursive

Flexible

User specifies "degree" of flexibility

System "sets" the glue to do the typesetting
    nontrivial algorithms

Sample .tex File

```
\hsize3in\vsize4.5in\nopagenumbers\vskip.5in
\centerline{\bf Manipulating polynomials}
\bigskip\noindent
Polynomial multiplication is easily accomplished
with term-by-term multiplication:
  $$(x^0+x^2)(x^3+x^5) = x^{ 3}+ 2x^{ 5}+x^{ 7}$$
but it could be even more easily accomplished if a
program were to do it.  The program should collect
terms, otherwise it might produce, for example,
$x^3+x^5+x^5+x^7$ as the expanded version of the
product $(x^0+x^2)(x^3+x^5)$; and it should
eliminate terms with 0 coefficients, otherwise it
might create extraneous output for an example like
  $$(x^1-x^2)(x^1+x^2) = x^{ 2}-x^{ 4}.$$
There are plenty of other conditions that need
to be checked. We could also differentiate:
$${d\over dx}(x^ 3+ 2x^5) = 3x^2 + 10x^4;$$
or integrate:
$$\int(x^3+2x^5) = {x^4\over4} + {x^6\over3}.$$
These are some of the situations that a program
for symbolically manipulating polynomials might
have to handle.
\vfill\eject\bye
```

Sample TeX output

Result of

tex sample.tex

lpr -d sample.dvi

Perspective on math applications

Use maple, matlab or mathematica, regularly!
- linear algebra
- elementary calculus
- statistics
- power series
- number theory
- linear programming (simplex)
- orthogonal polynomials
- ...

Indispensible tools in applied mathematics
- in widespread use
- can't be effective without such software

"Handbook of Math Functions" (and much more)
- now encoded in such software.

System development = math research
- Ex: There exists an algorithm for integration

Can "write programs" in Maple/Mathematica
- same basic control primitives as C
- hard to notice difference in small programs
- stay within environment
- *but* switch to C if performance-critical

ISSUE: Are "hard problems" solved properly?
- still have to understand the computation

Use TeX to communicate results
- (.tex, .dvi, .ps, or .pdf on web)