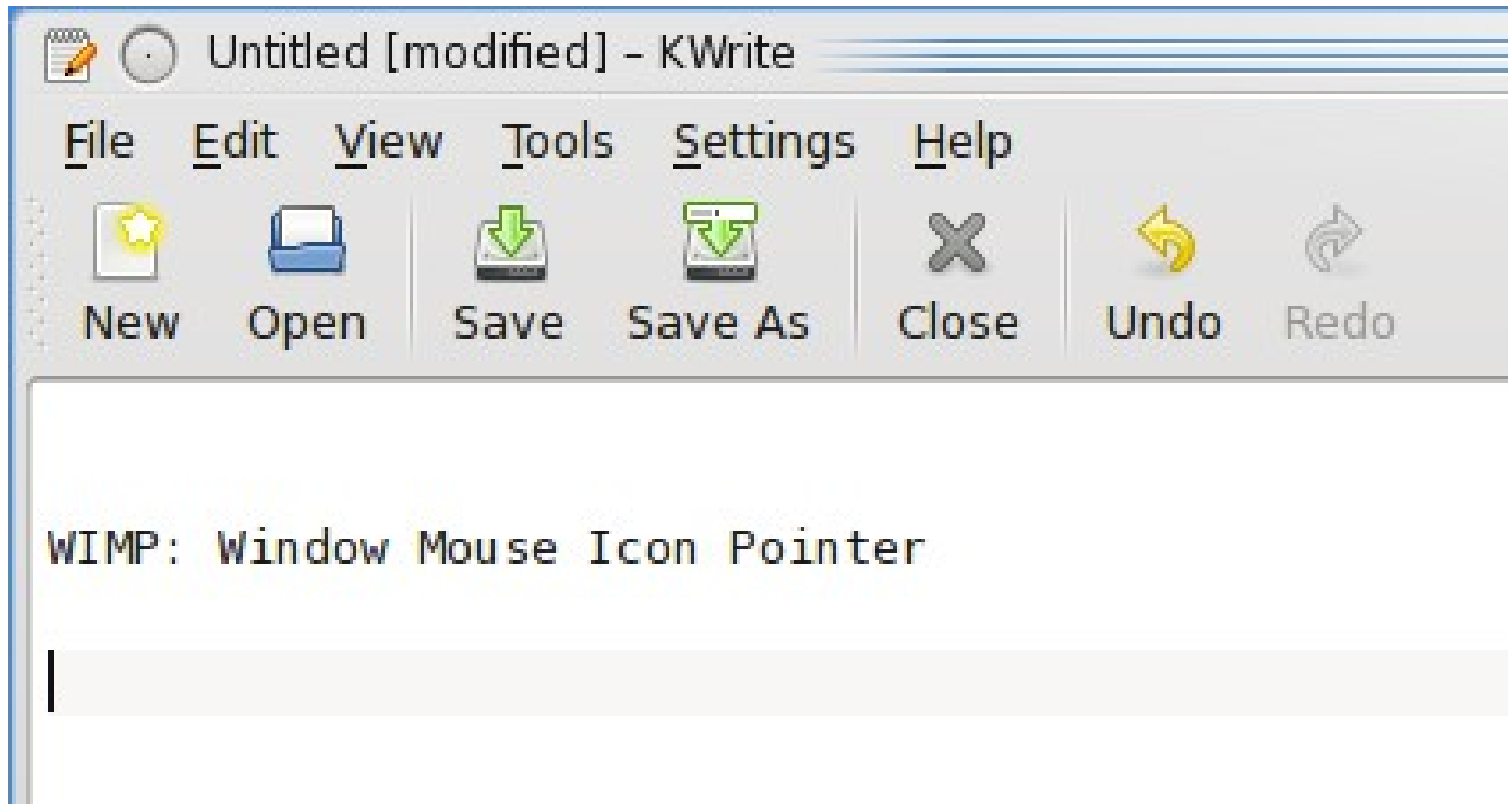
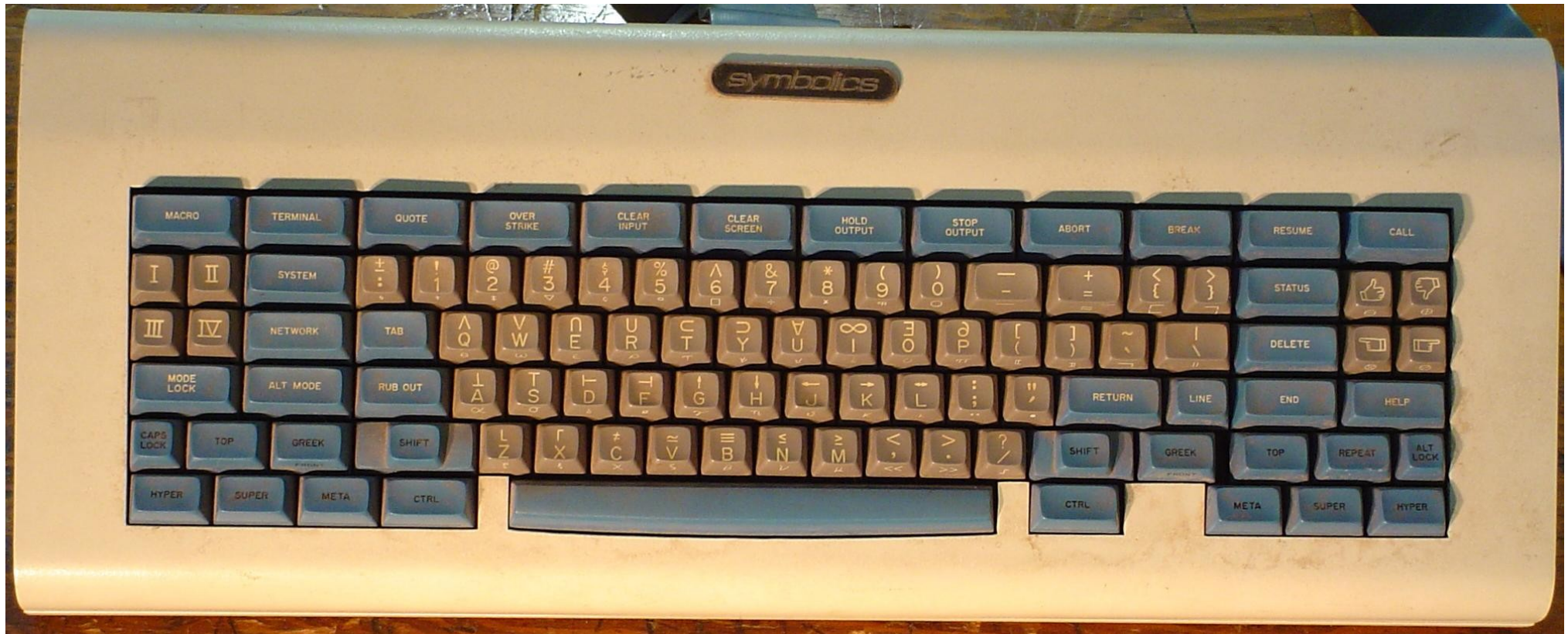


Text Editors ...



~ WYSIWYG

Who needs a mouse anyway ... “space cadet” keyboard



Who needs a mouse anyway ... the vi editor (and variants ... vim)

```
cc1plus
Auto
//
// Modified by S Zeil, Old Dominion University, adding code
// to read & write the question trees from a file.
//
//
# include <iostream.h>
# include <string>

#include "node.h"
#include "treeio.h"

bool answer() {
    // get yes no answer
    while (1) {
        string ans;
        getline(cin, ans);
        if ((ans[0] == 'y') || (ans[0] == 'Y'))
            return true;
        else if ((ans[0] == 'n') || (ans[0] == 'N'))
            return false;
        gout << "please answer yes or no.\n";
    }
}

(7 of 10): 'gout' undeclared (first use this function) 32,3-17 7%
```

Who needs a mouse anyway ... the vi editor

```
root@fedora:~  
File Edit View Terminal Tabs Help  
[root@fedora ~]# cat /etc/inittab  
# inittab is only used by upstart for the default runlevel  
#  
# ADDING OTHER CONFIGURATION HERE WILL HAVE NO EFFECT ON  
#  
# System initialization is started by /etc/event.d/rcS  
#  
# Individual runlevels are started by /etc/event.d/rc[0-6]  
#  
# Ctrl-Alt-Delete is handled by /etc/event.d/control-alt-del  
#  
# Terminal gettys (tty[1-6]) are handled by /etc/event.d/  
# /etc/event.d/serial  
#  
# For information on how to write upstart event handlers  
# upstart works, see init(8), initctl(8), and events(5)  
#  
# Default runlevel. The runlevels used are:  
# 0 - halt (Do NOT set initdefault to this)  
# 1 - Single user mode  
# 2 - Multiuser, without NFS (The same as 3, if you do not have networking)  
# 3 - Full multiuser mode  
# 4 - unused  
# 5 - X11  
# 6 - reboot (Do NOT set initdefault to this)  
#  
id:5:initdefault:  
[root@fedora ~]#
```



```
parport0: PC-style at 0x378, irq 7 [PCSP,TRISTATE]  
lp0: using parport0 (interrupt-driven).  
lp0: console ready  
Capability LSM initialized  
Checking non-root filesystems:  
fsck 1.39 (29-May-2006)  
usbfs on /proc/bus/usb type usbfs (rw)  
Mounting non-root local filesystems:  
nothing was mounted  
Using /etc/random-seed to initialize /dev/urandom.  
Skrypt startowy S zakonczony. Wszystko OK!  
INIT: Entering runlevel: 3  
Going multiuser...  
Updating shared library links: /sbin/ldconfig &  
Starting syslogd daemons: /usr/sbin/syslogd /usr/sbin/klogd -c 3 -x  
Triggering udev events: /sbin/udevtrigger --retry-failed  
Starting Internet super-server daemon: /usr/sbin/inetd  
Starting ACPI daemon: /usr/sbin/acpid  
Loading /usr/share/kbd/keymaps/i386/qwerty/pl.map.gz  
Starting gpm: /usr/sbin/gpm -m /dev/mouse -t ps2  
  
Welcome to Linux 2.6.21.5-smp (tty1)  
slack login: _
```

Skrypty startowe "rc.S" i "rc.3"

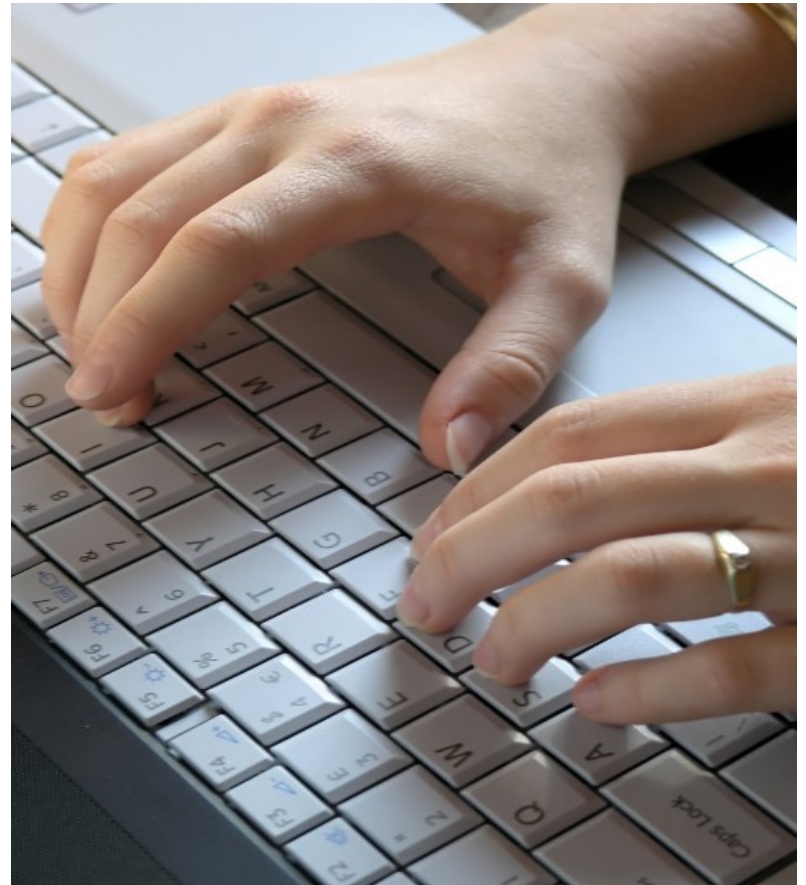
use in minimal environments, ubiquitous

Who needs a mouse anyway ...
the **vi** editor (based on **ex** line editor)

Two modes:

- Command
- Insert

Keyboard only!
... **fast** ...



vi help sheet

version of 29^o Oct. 2006

some useful tips and command for the vi editor

You have to use **ESC** (escape key) to get into the vi command mode. You will need to press **RETURN** Key for executing a command starting with the punctuation character " : " or " / " or " ? ". Use **CTRL+c** for cancel a command. Use " ." for repeat your last command and " !:cmd " for execute a shell command (where cmd is the command to execute).

to get into insert mode

i	insert text before the cursor
a	append text after the cursor
I	insert text at the beginning of the current line
A	append text at the end of the current line
o	insert text in a new line below the cursor
O	insert text in a new line below the cursor

navigation commands

h or ←	move cursor left
l or →	move cursor right
k or ↑	move cursor up
j or ↓	move cursor down
G	goto the end of file
nG or :n	goto the line number " n "
o	move to the beginning of line
\$	move to the end of line
ctrl+f	move one screen forward
ctrl+b	move one screen backward

file commands

:q	quit current open file
:q!	force to quit (without saving open file)
:w	save file
:w file	save file as " name "
:wq!	overwrite file then quit
:x,yw file	write from line " x " to line " y " into " file "
:w >> file	append buffer to " file "
:e file	edit another file
:e! file	edit another file without saving the current open file
:r file	insert file content at the current cursor position
:n	edit next file in vi arguments file list
ctrl+G	get file status

You can launch vi with some arguments. One usefull is " -c " for execute vi commands in a file directly from the command line.

ex. :
vi -c " %s/false/true/g|:wq " file.txt

search commands

/string	search forward for " string "
?string	search backward for " string "
n	repeat last search
:%s/str1/str2/gc	search and replace " str1 " by " str2 " from the current line (a line number can be specified before the comma) to the end of file. Ask for confirmation before replace.
:%s/str1/str2/g	replace all " str1 " by " str2 " in all the file without confirmation.
:%s/str1/str2/	replace " str1 " by " str2 " for first occurrence of each line of file.

edit commands

r	replace a character at the cursor position
u	undo last change
MA	set mark " A " (can be any letter, case sensitive) at the beginning of current line
y'A	yank from current line to the mark " A "
d'A	delete from current line to the mark " A "
P (caps P)	put the buffer before the cursor
p (small p)	put the buffer after the cursor
x	delete character at cursor position
dw	delete first word after cursor position
d\$ or D	delete from cursor position to the end of line
dd	delete curent line
J (caps J)	join curent line with the following line

Look at :
:map for mapping a key in command mode to a group of commands (ex. :map de :,\$d^M will delete all file when using de command)
:set for define or show your editor current options
:ab for define a text abbreviation in insert mode (ex. :ab VIM Vi Improved will auto complete VIM in insert mode)