

Unix Philosophy, Text Editors, IDEs

Comp Sci 1585
Data Structures Lab:
Tools for Computer Scientists



Unix
philosophy
What is text?
X-forwarding
Text editors
nano
Vim
Emacs
Kate
Atom
Integrated
Development
Environments
Code::Blocks
Building /
Running
Writing code
Qt creator
KDevelop

Unix
 philosophy

What is text?

X-forwarding

Text editors

nano
 Vim
 Emacs
 Kate
 Atom

Integrated
 Development
 Environments

Code::Blocks
 Building /
 Running
 Writing code
 Qt creator
 KDevelop

① Unix philosophy

② What is text?

③ X-forwarding

④ Text editors

nano

Vim

Emacs

Kate

Atom

⑤ Integrated Development Environments

Code::Blocks

Building / Running

Writing code

Qt creator

KDevelop

Unix
philosophy

What is text?

X-forwarding

Text editors

nano

Vim

Emacs

Kate

Atom

Integrated

Development

Environments

Code::Blocks

Building /

Running

Writing code

Qt creator

KDevelop

- Write programs to handle text streams, because that is a universal interface.
- Combine “small, sharp tools” and the use of “common underlying formatthe line-oriented, plain text file” to accomplish larger tasks.
- Store data in flat text files

https://en.wikipedia.org/wiki/Unix_philosophy

Unix
philosophy

What is text?

X-forwarding

Text editors

nano

Vim

Emacs

Kate

Atom

Integrated
Development
Environments

Code::Blocks

Building /
Running

Writing code

Qt creator

KDevelop

Another major tenet of the philosophy is to use plain text (i.e., human readable alphanumeric characters) rather than binary files (which are not fully human readable) to the extent possible for the inputs and outputs of programs and for configuration files. This is because plain text is a universal interface; that is, it can allow programs to easily interact with each other in the form of text outputs and inputs, in contrast to the difficulty that they would have if each used mutually incompatible binary formats and because such files can be easily interfaced with humans. The latter means that it is easy for humans to study, correct, improve and extend such files as well as to port (i.e., modify) them to new platforms (i.e., other combinations of operating systems and hardware).

Excerpts from the UNIX philosophy

Unix
philosophy

What is text?

X-forwarding

Text editors

nano

Vim

Emacs

Kate

Atom

Integrated
Development
Environments

Code::Blocks

Building /
Running

Writing code

Qt creator

KDevelop

Unix tradition strongly encourages writing programs that read and write simple, textual, stream-oriented, device-independent formats. Under classic Unix, as many programs as possible are written as simple filters, which take a simple text stream on input and process it into another simple text stream on output. Despite popular mythology, this practice is favored not because Unix programmers hate graphical user interfaces. It's because if you don't write programs that accept and emit simple text streams, it's much more difficult to hook the programs together. Text streams are to Unix tools as messages are to objects in an object-oriented setting. The simplicity of the text-stream interface enforces the encapsulation of the tools. More elaborate forms of inter-process communication, such as remote procedure calls, show a tendency to involve programs with each others' internals too much. To make programs composable, make them independent. A program on one end of a text stream should care as little as possible about the program on the other end. It should be made easy to replace one end with a completely different implementation without disturbing the other.

Unix
philosophy

What is text?

X-forwarding

Text editors

nano
Vim
Emacs
Kate
Atom

Integrated
Development
Environments

Code::Blocks
Building /
Running
Writing code
Qt creator
KDevelop

① Unix philosophy

② What is text?

③ X-forwarding

④ Text editors

nano

Vim

Emacs

Kate

Atom

⑤ Integrated Development Environments

Code::Blocks

Building / Running

Writing code

Qt creator

KDevelop

Unix
 philosophy

What is text?

X-forwarding

Text editors

nano
 Vim
 Emacs
 Kate
 Atom

Integrated
 Development
 Environments

Code::Blocks
 Building /
 Running
 Writing code
 Qt creator
 KDevelop

- 'Plain text' is ASCII-encoded data. (Or UTF-8 encoded data, or EBCDIC encoded data. . .)
- Most programming languages and network protocols use plain text because it is easy for developers to understand.
- Text editors are just tools for editing plain text files. You'll be using one a lot, so you should know how to use your editor of choice.

Unix
philosophy

What is text?

X-forwarding

Text editors

nano

Vim

Emacs

Kate

Atom

Integrated

Development

Environments

Code::Blocks

Building /
Running

Writing code

Qt creator

KDevelop

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Unix
philosophy

What is text?

X-forwarding

Text editors

nano
Vim
Emacs
Kate
Atom

Integrated
Development
Environments

Code::Blocks
Building /
Running
Writing code
Qt creator
KDevelop

You are responsible for submitting all text and source files you will submit in the entire class encoded UTF-8, Unix delimited.

Unix
philosophy

What is text?

X-forwarding

Text editors

nano
Vim
Emacs
Kate
Atom

Integrated
Development
Environments

Code::Blocks
Building /
Running
Writing code
Qt creator
KDevelop

① Unix philosophy

② What is text?

③ X-forwarding

④ Text editors

nano

Vim

Emacs

Kate

Atom

⑤ Integrated Development Environments

Code::Blocks

Building / Running

Writing code

Qt creator

KDevelop

- X-windows is Linux's system for displaying graphical programs.
- X server: Program that manages what is displayed on the screen.
- X client: Program that wants to display something.
- Windows: Run Xming to start the server, then use PuTTY like normal.
- Mac: Install X11.app, then use `$ ssh -X <hostname>`
- Linux: You are already running an X server! Just use `$ ssh -X <hostname>`
- Use `$ xeyes` to check if X forwarding is set up correctly.

Unix
philosophy

What is text?

X-forwarding

Text editors

nano
Vim
Emacs
Kate
Atom

Integrated
Development
Environments

Code::Blocks
Building /
Running
Writing code
Qt creator
KDevelop

① Unix philosophy

② What is text?

③ X-forwarding

④ Text editors

nano

Vim

Emacs

Kate

Atom

⑤ Integrated Development Environments

Code::Blocks

Building / Running

Writing code

Qt creator

KDevelop

Unix
 philosophy
 What is text?
 X-forwarding
 Text editors
nano
 Vim
 Emacs
 Kate
 Atom
 Integrated
 Development
 Environments
 Code::Blocks
 Building /
 Running
 Writing code
 Qt creator
 KDevelop

① Unix philosophy

② What is text?

③ X-forwarding

④ Text editors

nano

Vim

Emacs

Kate

Atom

⑤ Integrated Development Environments

Code::Blocks

Building / Running

Writing code

Qt creator

KDevelop

Unix
philosophy

What is text?

X-forwarding

Text editors

nano

Vim

Emacs

Kate

Atom

Integrated

Development

Environments

Code::Blocks

Building /
Running

Writing code

Qt creator

KDevelop

First part of the command:

- $\hat{-}$ is `Ctrl`
- M- is `Alt`

Full command examples:

- "Write out" (save): $\hat{-}$ O
- Quit: $\hat{-}$ X
- Undo: M-U
- Redo: M-E

Most importantly:

- Help: $\hat{-}$ G

Unix
 philosophy
 What is text?
 X-forwarding
 Text editors
 nano
Vim
 Emacs
 Kate
 Atom
 Integrated
 Development
 Environments
 Code::Blocks
 Building /
 Running
 Writing code
 Qt creator
 KDevelop

1 Unix philosophy

2 What is text?

3 X-forwarding

4 Text editors

nano

Vim

Emacs

Kate

Atom

5 Integrated Development Environments

Code::Blocks

Building / Running

Writing code

Qt creator

KDevelop

Unix
philosophy

What is text?

X-forwarding

Text editors

nano

Vim

Emacs

Kate

Atom

Integrated

Development

Environments

Code::Blocks

Building /
Running

Writing code

Qt creator

KDevelop

- Buffer: Opened file
- Window: Visual section displaying a buffer
- (Atom borrowed this terminology from vim!)
- Composability: You can connect simple commands together into complex ones
- `$ vimtutor` can teach you some basic vim stuff

Tip: `$:set mouse=a`

Modes: Keys do different things in different modes

Unix
philosophy

What is text?

X-forwarding

Text editors

nano

Vim

Emacs

Kate

Atom

Integrated

Development

Environments

Code::Blocks

Building /
Running

Writing code

Qt creator

KDevelop

- Normal: Navigation/commands. `Esc`, `Ctrl` + `c`
- Insert: Writing text.
 - `i`: Insert at cursor
 - `I`: Insert at beginning of line
 - `a`: Insert after cursor (append at cursor)
 - `A`: Insert at end of line (Append to line)
 - `o`: Insert on new line below cursor
 - `O`: Insert on new line above cursor
 - `c`: Change text at cursor
- Replace: Overwriting text. `R`
- Visual: Selecting text.
 - `v`: Character select
 - `V`: Line select
 - `Ctrl` + `v`: Block select

Unix
philosophy

What is text?

X-forwarding

Text editors

nano
Vim
Emacs
Kate
Atom

Integrated
Development
Environments

Code::Blocks
Building /
Running
Writing code
Qt creator
KDevelop

- `j`/`k`/`h`/`l`: up/down/left/right
- `^`/`$`: Beginning/end of line
- `w`/`e`/`b`: Next word/Next word end/Back one word
- `%`: Matching brace or parenthesis
- `gg`/`G`: Top/bottom of document

Tip: Repeat commands by specifying a number first: `$ 2w`

Tip: Compose with insert modes: `$ c4w`

- `:w` Save
- `:q` Quit
- `:wq`/`ZZ` Save and quit
- `u`/`Ctrl+r`: undo/redo
- **Clipboards**
 - `y`/`yy`: Copy/'yank' (line)
 - `d`/`dd`: Cut/'delete' (line)
 - `p`/`P`: Paste after cursor/Paste before cursor
 - `" + y`: Yank to system clipboard
- `>`/`<`: Indent/Outdent
- `=`: Autoformat

Unix
philosophy

What is text?

X-forwarding

Text editors

nano

Vim

Emacs

Kate

Atom

Integrated
Development
Environments

Code::Blocks

Building /
Running

Writing code

Qt creator

KDevelop

Configuration:

- `$.vimrc` in your home directory
- Example config on the website

Plugins:

- Airline: Pretty status bar
- Fugitive: Git integration
- Syntastic: Syntax checking

Links:

- [Vim Cheat Sheet](#)
- [Another cheat sheet](#)

Unix
 philosophy
 What is text?
 X-forwarding
 Text editors
 nano
 Vim
Emacs
 Kate
 Atom
 Integrated
 Development
 Environments
 Code::Blocks
 Building /
 Running
 Writing code
 Qt creator
 KDevelop

① Unix philosophy

② What is text?

③ X-forwarding

④ Text editors

nano

Vim

Emacs

Kate

Atom

⑤ Integrated Development Environments

Code::Blocks

Building / Running

Writing code

Qt creator

KDevelop

Unix
 philosophy

What is text?

X-forwarding

Text editors

nano

Vim

Emacs

Kate

Atom

Integrated

Development

Environments

Code::Blocks

Building /

Running

Writing code

Qt creator

KDevelop

Use `$ emacs -nw` to launch in the terminal (instead of a GUI)

- Frame: Everything you can see (you can have multiple frames!)
- Buffer, Window: As with vim and atom
- Instead of `Ctrl + n`, emacs writes this as `$ C-n`
- Meta key (written `$ M-x`): Use `Alt + x` or `Esc x`
- In Emacs: `$ C-h t` starts a tutorial

Unix
 philosophy

What is text?

X-forwarding

Text editors

nano
 Vim
Emacs
 Kate
 Atom

Integrated
 Development
 Environments

Code::Blocks
 Building /
 Running
 Writing code
 Qt creator
 KDevelop

- `$ C-n` / `$ C-p` next/previous line
- `$ C-f` / `$ C-b` forward/back one character
- `$ M-f` / `$ M-b` forward/back one word
- `$ C-a` / `$ C-e` Beginning/end of line
- `$ C-v` / `$ M-v` Down/Up one page

Unix
 philosophy

What is text?

X-forwarding

Text editors

nano
 Vim
Emacs
 Kate
 Atom

Integrated
 Development
 Environments

Code::Blocks
 Building /
 Running
 Writing code
 Qt creator
 KDevelop

- Deleting: Backspace(`$ `)/ `$ C-d`
- Kill (cut) a word: `$ M-` / `$ M-d`
- Kill to the end of a line: `$ C-k`
- Unkill (paste): `$ C-y`
- `$ M-y` to cycle through kills

Unix
 philosophy

What is text?

X-forwarding

Text editors

nano
 Vim
Emacs
 Kate
 Atom

Integrated
 Development
 Environments

Code::Blocks
 Building /
 Running
 Writing code
 Qt creator
 KDevelop

- Saving a file: `$ C-x C-s`
- Quitting: `$ C-x C-c`
- Stop a command: `$ C-g` or hit `Esc` three times
- Undo: `$ C-_`
- Windows:
 - Make 2 windows: `$ C-x 2`
 - Jump between windows: `$ C-x o`
- Links:
 - Reference Card
 - Emacs Tour

Unix
 philosophy
 What is text?
 X-forwarding
 Text editors
 nano
 Vim
 Emacs
Kate
 Atom
 Integrated
 Development
 Environments
 Code::Blocks
 Building /
 Running
 Writing code
 Qt creator
 KDevelop

① Unix philosophy

② What is text?

③ X-forwarding

④ **Text editors**

nano

Vim

Emacs

Kate

Atom

⑤ **Integrated Development Environments**

Code::Blocks

Building / Running

Writing code

Qt creator

KDevelop

Unix
philosophy

What is text?

X-forwarding

Text editors

nano
 Vim
 Emacs
Kate
 Atom

Integrated
 Development
 Environments

Code::Blocks
 Building /
 Running
 Writing code
 Qt creator
 KDevelop

- Linux-only
- Can be used as a simple text editor with no learning curve
- Can also be used in Vi/m input mode
- Has many advanced features
- Like other KDE applications, is highly configurable

Unix
 philosophy
 What is text?
 X-forwarding
 Text editors
 nano
 Vim
 Emacs
 Kate
Atom
 Integrated
 Development
 Environments
 Code::Blocks
 Building /
 Running
 Writing code
 Qt creator
 KDevelop

① Unix philosophy

② What is text?

③ X-forwarding

④ Text editors

nano

Vim

Emacs

Kate

Atom

⑤ Integrated Development Environments













Code::Blocks

Building / Running

Writing code

Qt creator

KDevelop

- Terminology
 - Buffer: Contents of an open file.
 - Pane: A visual section of the editor. Holds buffers, status bars, etc.
- Shortcuts
 -  +  +  Show a directory in the tree view
 -  +  +  Command palette you can search through
 - Searching for files:
 -  + ,  +  Find a file in the current open directories
 -  +  Jump to an open buffer
- Packages
 - **minimap** Shows a scrollable overview of your file
 - **autocomplete-clang** C++ Autocompletion
 - **vim-mode** Some vim shortcuts
- Link (click following): [Documentation](#)

Unix
philosophy

What is text?

X-forwarding

Text editors

nano
Vim
Emacs
Kate
Atom

Integrated
Development
Environments

Code::Blocks
Building /
Running
Writing code
Qt creator
KDevelop

① Unix philosophy

② What is text?

③ X-forwarding

④ Text editors

nano

Vim

Emacs

Kate

Atom

⑤ Integrated Development Environments

Code::Blocks

Building / Running

Writing code

Qt creator

KDevelop

What are Integrated Development Environments?

Unix
philosophy

What is text?

X-forwarding

Text editors

nano
Vim
Emacs
Kate
Atom

Integrated
Development
Environments

Code::Blocks
Building /
Running
Writing code
Qt creator
KDevelop

- Commonly referred to as an IDE
- Combines editor, project management, compiler, debugger, etc. into one program.
- Commonly used on Windows (and OS X, to a lesser extent).
- Usually feature code completion.
- Provide a build management system for complicated projects.
- Some come with graphical tools for building GUIs.

Unix
 philosophy
 What is text?
 X-forwarding
 Text editors
 nano
 Vim
 Emacs
 Kate
 Atom
 Integrated
 Development
 Environments
Code::Blocks
 Building /
 Running
 Writing code
 Qt creator
 KDevelop

① Unix philosophy

② What is text?

③ X-forwarding

④ Text editors

nano

Vim

Emacs

Kate

Atom

⑤ Integrated Development Environments

Code::Blocks

Building / Running

Writing code

Qt creator

KDevelop

Unix
philosophy

What is text?

X-forwarding

Text editors

nano
Vim
Emacs
Kate
Atom

Integrated
Development
Environments

Code::Blocks
Building /
Running
Writing code
Qt creator
KDevelop

- `f9` Build and run
- `Ctrl + f9` Build
- `Ctrl + f10` Run
- `Project >> Build Options` Enable `$ -Wall`

- `Ctrl` + `.` Go to function implementation.
- `Ctrl` + `↑` + `.` Go to function declaration.
- `Ctrl` + `space` Show completions.
- Right-click on a file and choose 'Format this file' to autoformat.

Unix
 philosophy
 What is text?
 X-forwarding
 Text editors
 nano
 Vim
 Emacs
 Kate
 Atom
 Integrated
 Development
 Environments
 Code::Blocks
 Building /
 Running
 Writing code
Qt creator
 KDevelop

- ① Unix philosophy
- ② What is text?
- ③ X-forwarding
- ④ Text editors
 - nano
 - Vim
 - Emacs
 - Kate
 - Atom
- ⑤ Integrated Development Environments
 - Code::Blocks
 - Building / Running
 - Writing code
 - Qt creator**
 - KDevelop

Unix
 philosophy

What is text?

X-forwarding

Text editors

nano
 Vim
 Emacs
 Kate
 Atom

Integrated
 Development
 Environments

Code::Blocks
 Building /
 Running
 Writing code
 Qt creator
 KDevelop

- Excellent general purpose C++ IDE that is also used for Qt-GUI development.
- Requires that you create a “project” rather than just running .cpp files quickly.
- Excellent debugger (GDB) integration and visualization (more to come later).

Unix
 philosophy
 What is text?
 X-forwarding
 Text editors
 nano
 Vim
 Emacs
 Kate
 Atom
 Integrated
 Development
 Environments
 Code::Blocks
 Building /
 Running
 Writing code
 Qt creator
 KDevelop

1 Unix philosophy

2 What is text?

3 X-forwarding

4 Text editors

nano

Vim

Emacs

Kate

Atom

5 Integrated Development Environments

Code::Blocks

Building / Running

Writing code

Qt creator

KDevelop

Unix
philosophy

What is text?

X-forwarding

Text editors

nano
 Vim
 Emacs
 Kate
 Atom

Integrated
 Development
 Environments

Code::Blocks
 Building /
 Running
 Writing code
 Qt creator
KDevelop

- IDE natively intended for C++/C, but that can be used for other languages like Python as well.
- Requires that you create a “project” rather than just running .cpp files quickly.
- Pretty good debugger (GDB) integration and visualization.