

lock

address

flags

frame

page

path

mask

root

spin

atomic

key

trace

queue

clocks

table

raw

head

stream

map

handler

ring

tail

self

callback

stack

dig

pipe

thread

watchdog

scratch

unlikely

magic

vortex

bucket

pool

loop

quirk

job

owner

leaf

mailbox

prism

flush

family

diva

words

filter

chain

bat

fence

dirty

cookie

bail

tunnel

journal

window

ether

dead

trees

walk

voice

hook

slave

dummy

fabric

dropped

nested

signature

slice

silence

listen

doorbell

cast

semaphore

palette

fat

heap

topology

roles

orphan

markers

migrate

tea

mirror

room

trap

heartbeat

traffic

savage

identity

generation

possible

relax

waiter

advertise

digest

forks

buddy

wrap

spare

smack

daemon

stage

freezer

sink

flock

native

suite

handshake

stale

neighbour

elevator

eviction

nibble

fire

swim

flight

sacked

inject

disposition

actor

writer

loud

barker

undecorated

grace

lease

glue

beg

sharpness

diversity

stolen

dwarf

flavor

visible

wheel

bipolar

fold

maybe

balloon

poison

dark

trailer

affinity

garbage

stuff

conflict

congestion

zeros

desire

leader

form

cork

unwritten

nothing

drift

leak

ladder

sigh

loft

yeah

universe

boxes

promise

collector

karma

lint

corruption

watcher

scissor

picture

leftover

zombie

silly

taint

storm

graveyard

bra

sleeper

tie

knot

mess

linger

trampoline

touched

cows

interpreter

bridges

gang

coexistence

corkscrew

daylight

factory

openers

toe

why

ceiling

letter

litter

rope

bakery

landscape

spool

rubber

martians

mastery

pretty

dubious

coffee

surplus

vote

nowhere

crap

tickets

optimistic

balancer

dada

haystack

poodle

mind

hangover

backdoor

skinny

refrigerator

needle

lie

gesture

canary

primitives

thing

rudder

eaten

skeleton

laundromat

launder

hose

breadcrumb

withdrawal

art

lack

epilogue

agnostic

## METHODOLOGY

These words were extracted from the source code of the Linux kernel. In computing, a kernel is the core component of an operating system, managing the system's resources and enabling all other software to run.

In the process of extraction, string literals ("like this") were discarded as they contain messages geared towards users rather than internal concepts. Also discarded were comments in the code as its prose can only be inadequately captured in a word list. Finally, the keywords of the programming language itself were ignored, as they are preset and fixed, and thus don't inform us about the specific software we're looking at.

What is left are: custom variable names, custom data types, function names and defines. All of which can, in theory, be arbitrarily named without impacting the

software's functionality. Yet since the men and women writing these programs have to be intimately familiar with these names to be able to change the program's behavior, great care is taken to pick a name that best represents its content.

In other words, if I were to model an abstract piece of code based on my knowledge of how an escalator works - I would likely name it "escalator" (with a function "call", for example).

The preceding list of words is exactly this: a heavily curated record of some of the more unexpected and poetic concepts that are at work whenever we boot our machines.





MARCH 2012

GOTTFRIED HAIDER