

Lars Wirzenius <liw@iki.fi>

Programmer survival skills

How to avoid unemployment, starvation, and ennui

Jyväskylä, December 8, 2005

Programming: what is it?

- Programming is the automatization of data processing tasks
 - economic restrictions
 - technical restrictions
 - following specifications
 - quality
- Software engineering = performing programming in the hope of making money

Don't believe all the crap you're fed

- Programming is the most fun thing you can do alone that humanity has come up with
 - An art form, support for megalomania, an ego building closed positive feedback loop, easiest way to reach Nirvana ...
- If programming isn't fun, you're in the wrong place
 - There are less onerous ways to make a living

Fun = hack mode

- "Flow"
 - state of deep concentration
 - requires at least 15 minutes of calm to start
 - easily broken by disturbances
- You can do without, but time requirements grow many times bigger
 - and it's not at all as fun
 - hack mode = nirvana

Things are not always fun...

- Making a living is not always fun
- Somebody else ... tells you what to do
 - how to do it
 - is often clueless
 - interrupts and disturbs all the time
 - and goes bankrupt
- Luckily you can program on something fun in your free time

Choosing an employer

- Start ups, big companies, being a consultant...
 - Figure out what suits you and stick to that
 - Experiment before you have family and a mortgage
- Do they understand what aids productivity?
 - Do they have calm work spaces, ergonomics, ...
- Is the job contract sensible?
 - Read laws, use a lawyer (in Finland, unions have 'em)
 - Copyrights/patents vs stuff done on your free time

Managing your work environment

- Noise
 - GOOD earphones (worn 8+ hours/day)
- Interruptions
 - Your own room, close the door
 - Laptop + working at home
- Ergonomics
 - Chairs, desks, reflections from the screen, ...
 - Employer's legal responsibility!

Workspace: good



Workspace: not
so good



Workspace: typical (and bad)



Make your environment interesting

- Toys?
- Books?
- Magazines?
- Pics, comics, copier art on walls?
- Doodling place? Whiteboard, paper, ...
- Boxing bag?
 - Or other way of letting out steam
- Coffee room?

Practical skill #1

- Touch typing!
 - Or anything that works
- Megabytes / year
- Easy to learn by yourself, even

QWERTY? Dvorak? Who cares, use whatever you want.



Practical skill #2

- Participating in or chairing meetings
- **CONCENTRATE ON ESSENTIALS**
 - Don't meander, skip irrelevancies, don't gossip, ...
 - PREPARE!
- Notes/minutes are a good thing. Really.
 - Even if just a few key words
- Avoid unnecessary meetings

Practical skill #3

- Time management
 - Do all things happen in time?
 - Do you know beforehand when things will be done?
 - You're always going to be busy
 - You're always going to have too much to do
- "No", "there's not enough time", and "in your dreams" are good phrases to learn
- Many systems and methods, lots of opportunity for fun optimization, tweaking, tuning, ...

These are important too

- An ability to write non-fiction well
 - Clearly, accurately, quickly, briefly
 - Expressing yourself requires practice
- An ability to use graphs/pictures to express things
 - A small amount of statistics is good for anyone
- Communication skills in general
 - discussions, e-mail, giving presentations, ...
 - anyone can learn, just takes some practice

Programming skills: top 3

- Avoiding, finding, fixing bugs
 - designing test cases, debugging tactics, ...
- Finding information (google, wikipedia, ...)
 - Oh yeah: libraries exist, too
- Quick learning, continuous study
 - Wrote COBOL for 30 years at the same job? Don't call us, we won't call you...

Difficult programming skills

- Empathy
 - What do people really want out of the software?
 - What's easy to them, and what's hard?
- Life, getting one
 - A sense of perspective is surprisingly rare and difficult to have
- Common sense, how to keep things simple
 - one million or ten million lines of code?

The most important languages for a programmer

- Native language (if not English)
- English
- Some programming languages too, obviously
 - Something imperative (C?)
 - Something object oriented (Python?)
 - Something functional (Lisp/Scheme/Haskell?)
 - Write at least one significant program in each

How to look like a good programmer?

- Put lots of popular acronyms in your CV
 - The HR dept. is a pattern matching engine
- Have programming as a hobby
 - Lots of projects of your own, even if in progress
 - Or participate in other people's projects, e.g., free software ones
- Know how to talk and give a good impression

How to become a good programmer?

- Write lots of program, also as a hobby
 - different kinds of programs
- Maintain (at least read) other people's code
- Read a lot in general
 - programming literature and other things
- Discuss, argue, think
 - big programs are social events
- Write about programming, give talks

You may need to understand hardware, too



Using LEDs for status output

- color = black, red, green
- three LEDs
- QUESTION: How many states can you show?



Which way is the box?



Answer: four states

- No lights
- One light
- Two lights
- Three lights

Reading list

- DeMarco, Tom; Lister, Timothy: Peopleware: productive projects and teams