

<u>Offset</u>	<u>Topic</u>
00:17	<ul style="list-style-type: none"> • Intro
	<ul style="list-style-type: none"> • 3rd anniversary
01:46	<ul style="list-style-type: none"> • Listener Feedback
	<ul style="list-style-type: none"> • CA on BrightKite <ul style="list-style-type: none"> • Good to see a service that gets at least a minimal structure to relationships • I think this may be more risk driven • It makes a bit more sense that way • There doesn't seem to be the same pressure on messaging • Other than the awkward moments that arise • Or the decision to block or not share • Paul with correction on white spaces <ul style="list-style-type: none"> • http://thecommandline.net/2008/06/15/news_144/#comment-431 • Paul with anecdote about unintended uses <ul style="list-style-type: none"> • http://thecommandline.net/2008/06/11/unintended_uses/#comment-430
09:10	<ul style="list-style-type: none"> • Word of the Week: deep magic
	<ul style="list-style-type: none"> • http://catb.org/jargon/html/D/deep-magic.html
10:12	<ul style="list-style-type: none"> • Inner Chapter: Hacker Work Ethic
	<ul style="list-style-type: none"> • Hacker ethic <ul style="list-style-type: none"> • http://en.wikipedia.org/wiki/Hacker_ethic • Reference's Levy's Hackers and Himanen's The Hacker Ethic • Plan to read both, will discuss when I do • Speaks more to what hackers, by and large, consider right • Quickly <ul style="list-style-type: none"> • Sharing • Openness • Decentralization or distribution authority, effort • Free access to computers • Some form of improvement, personal or societal • Do it yourself and recognition of technical accomplishment, merit • Want to talk about work ethic, though <ul style="list-style-type: none"> • More directly informs practice of hacking • Think there is some overlap • In particular in the effort to be recognized for work, doing for yourself • Himanen's work actually informs this more <ul style="list-style-type: none"> • Passion • Hard work

Offset

Topic

- Creativity
- Joy in creating
- Emphasis is on value of character rather than ethics as rules and consequences
- I think it can be simplified to some more applicable thoughts
- If you want to create it, do it just for the joy of it
 - Who cares if there are dozens of programs that do the same thing
 - Hacking is about learning by doing
 - If you haven't written that sort of program but want to learn from it, do it
 - No one says you have to publish or share your program
 - If you do decide to share, you may find your approach is different enough
 - You may also re-work your code as a contribution to a similar project
 - If you have more experience with a whole program, can make submitting a patch easier
 - May lead you to more specific questions for development lists
 - Why the designers, coders on an existing project did what they did
 - Will certainly equip you to better understand the answers
 - If there really are dozens of programs like yours, temper yourself, expectations
 - Others may see your effort as wasted, can be discouraging
 - Keep strongly in mind why you did it, especially if it was just to learn or for fun
 - Have had a couple of candidates in interviews miss this point
 - Super enthusiastic about writing their own "yet another" program
 - If they say they did it not expecting anyone to use, just to understand, good answer
 - If they really expect to compete with Apache or Vim or Lint, not so much
- If its is missing, write it yourself
 - When you need some bit of software and have trouble finding it
 - Nothing says you cannot fill your own need
 - This is easier for simpler needs
 - May not be undertaken as lightly for larger needs
 - For example, writing a quick and dirty file parser
 - Versus a multiple user, large scale server
 - Make sure you've exhausted reasonable venues
 - Search engines, common software sites like SourceForge
 - Don't reinvent the wheel unless you have to
 - Be sure that a close match cannot be adopted, find counter examples, deal breakers
- If its broken, fix it yourself

Offset

Topic

- If you find a bug on an open or free project, you have the sources
- You have permission to change the sources
- Understand the license, so you know your obligations if you change them
- There is no real reason you can't fix a bug
- All software is built of the same pieces at the lowest levels
- Only limitation is time
- Even knowledge can be thought of as the time needed to understand
- If your patch is not accepted, still worth the effort
- Projects vary on how active they are
- A lively own may respond quickly to your bug report
- Less so ones may never fix your bug
- A simple or single bug shouldn't prevent you from using otherwise good software
- If you started it, finish it
 - This is easy when building software for pay
 - Without an expectation of releasing or publishing, may be tempting to stop halfway through
 - Others will value your efforts more if most of them are finished
 - A resume or project list full of unfinished work doesn't recommend you, to a community or an employer
 - Occasionally there are valid reasons for not finishing
 - Unforeseen interrupts, like real life problems or changes
 - Changing over to contribute to an existing project
 - Continuity of effort can be just as good
 - Abandoned your own effort but transfer knowledge, skills, contribution to similar project
 - I've seen many published, but open or free, projects unfinished
 - If you are on the only contributor, may want to factor in scale, time involved
 - Same concerns when deciding to write to fill a gap
- Emphasis with all of these is on you, doing on your own
 - Self sufficiency is good
 - Have to be careful not to develop blinders
 - Good code is also developed by groups
 - Even sole contributors often build on what has come before
 - Too much focus on your own efforts can bias you against the work of others
 - Re-inventing the wheel
 - Writing code when there is something serviceable already
 - Not invented here
 - Discounting code for reasons other than its technical merit

Offset

Topic

- The language choice, coding style or even personality of the author
- I've seen less emphasis on coordinating groups
 - Functions of traditional project management
 - That should not be so
 - Usually when we see it, it is for a "benevolent dictator", like Linus Torvalds
 - Critical for larger projects
 - Most popular, useful programs are of considerable scale
 - Linux itself, most of the internet servers, desktop environments, distributions
 - Probably seen as necessary evil
 - Is not code related, hard to quantify in terms of hard benefit
 - Direct project contributors are more likely to appreciate than users, occasional contributors
- Share what you've created
 - You don't always have to but better if you did
 - Can receive input, suggestions
 - Can earn a reputation based on your actual work
 - Can share in different ways
 - Can and should use supplemental documents, code comments to clarify why you are sharing
 - Nothing worse than someone mistaking sample code for production ready
 - Source code is the obvious choice
 - Not all source code needs to be released with the intent to use
 - Can often publish sources more as forms of examples
 - What you can do
 - How to solve particular coding problems
 - Just to highlight a particular creative bit of coding

32:29

• **Outro**

- Contact me
 - Email to feedback@thecommandline.net
 - Web site at <http://thecommandline.net/>
 - IM to [command.line@skype](skype:command.line)
 - Listener comment line is 240-949-2638
 - del.icio.us tag is "for:cmdln"
 - <http://twitter.com/cmdln>
- I'd like to thank libsyn.com for AAC hosting and Wouter de Bie for MP3 hosting
- These notes and the show audio and music are covered by a Creative Commons license
 - <http://creativecommons.org/licenses/by-nc-sa/3.0/us/>
 - Attribution, non-commercial, share alike