

2008-04-02 Show Notes

<u>Offset</u>	<u>Topic</u>
00:17	<ul style="list-style-type: none">• Intro
01:57	<ul style="list-style-type: none">• Listener Feedback<ul style="list-style-type: none">• [Si]dragon with book recommendation<ul style="list-style-type: none">• Link to audio from NPR interview<ul style="list-style-type: none">• http://podcastdownload.npr.org/anon.npr-podcasts/podcast/305/510071/88106931/WAMU_88106931.mp3• Richard Sennett, The Craftsman• Tor, the NRK doctrine<ul style="list-style-type: none">• http://feeds.feedburner.com/~r/nrkbeta/~3/258183916/• Jon on parallel computing<ul style="list-style-type: none">• http://www.eecs.berkeley.edu/Pubs/TechRpts/2006/EECS-2006-1.pdf• http://en.wikipedia.org/wiki/Dining_philosophers_problem• Multimode, Chromed Pork Podcast?<ul style="list-style-type: none">• http://www.chromedpork.net/• Would rather I not censor myself, deliver podcasts timely• Quantico Circuit at Verizon
12:06	<ul style="list-style-type: none">• Word of the Week: cycle<ul style="list-style-type: none">• http://catb.org/jargon/html/C/cycle.html
13:37	<ul style="list-style-type: none">• Hacking 101: Scripting Languages<ul style="list-style-type: none">• Scripting languages defined<ul style="list-style-type: none">• http://en.wikipedia.org/wiki/Scripting_language• Embedded within program• Used to control, or script, another program• So called for stage scripts, pre-set dialogue, actions for actors• Often fully interpreted or partially compiled• Means the program reads the same text as the hacker and executes existing functions• Byte code is marginally more efficient, saves text parsing• Still interpreted one operation at a time, rather than compiled to native instructions• Early script languages called batch or job control languages• Batch refers to batches of commands• Job control refers to being about to put a program in the background• On older systems where execution was slow, costly, saved operator time on repetitive tasks• Still a useful tool for reducing error, saving human time• Shell scripts<ul style="list-style-type: none">• Not embedded in a user program

Offset

Topic

- Used for scripting an operating system
- Shell refers to an interactive shell, or command line
- In Unix, Linux user can choose from a variety of shells with different features
- This shell choice also often affects the format, available commands for scripting
- Creator of Tcl, an older script language, defines dichotomy between system, scripting languages
 - http://en.wikipedia.org/wiki/Ousterhout%27s_dichotomy
 - Three criteria
 - Static vs. dynamic type
 - Complex vs. simple data structures
 - Compiled to machine code vs. interpreted or compiled to byte code or parse tree
 - Distinction is increasingly blurry
 - Java was originally interpreted, still often runs in mixed mode
 - Perl was originally interpreted
 - Perl 6 will run in a VM, Parrot, as byte code
 - Both now take advantage of JIT compilers
 - Byte code, VMs, open up interesting possibilities
 - Can write different parsers to generate same byte code
 - Jython in JVM, long list of prospective languages in Parrot
- Script languages independent of programs
 - Further breaks down Oustershout's dichotomy
 - Many languages have evolved with their own interpreters
 - Can be embedded in any number of programs
 - Saves a program author from creating own language, interpreter
 - Can be used across multiple programs
 - As independent languages, quickly acquired full language features, like variables, flow of control
 - Achieve Turing completeness, ability to express any algorithm that could be written in any other complete language
 - Retained interpreted nature, some have sophisticated interpreters like Parrot
 - Usually dynamic, consequence of roots, variables, type relied on what parent program exposes, provides
- Why are scripting languages important?
 - Quick and dirty tasks
 - Putting an existing program to new uses
 - Saves re-inventing functionality you need that may already exist
 - Automate tasks to reduce error
 - Alternative is to run programs manually
 - Too easy to make a mistake when running repeatedly

Offset

Topic

- Many macro languages originally developed to automate compilation
- Often first introduction to programming
 - Non-programmers may be comfortable scripting a program with which they are fluent
 - Learning curve for proto-hackers is shallow
 - Limited features, simpler syntax
 - Often very forgiving, quicker to make changes and re-run without a full compile
 - Some can be run in an interactive mode, like Ruby, Perl, shell
 - You can make a change, write a quick expression, get immediate feedback
- Standalone scripting languages
 - Low cost to entry for general programming
 - Widely, freely available
 - A lot of operating systems bundle the more popular ones, especially Perl
 - Easy enough to install on Windows
 - Microsoft has their own versions, not all are free, like VB, Visual Studio
 - Once you master a standalone script language, the gap to a system language is not wide
 - Learning the compiler may take some time, but most language concepts will translate
 - Some script languages have comparably complex tooling
 - Think Ruby on Rails which involved server, deployment
- Many environments rely heavily on scripting
 - Web browsers, web applications
 - Increasingly popular since they scrub out OS differences
 - This is the reason Microsoft competed so aggressively against Netscape in the 90's
 - Web browser involves multiple embedded languages
 - HTML for information, document structure
 - CSS for presentation, appearance
 - HTML, CSS are declarative, not scripting
 - JavaScript for behavior, drives an object model of the browser, DOM
 - Wholly interpreted
 - Is Turing complete, though
 - Web applications
 - Typically embed server execution instructions in document to be delivered to browser
 - Run in a web server program
 - Have access to the web server's environment

Offset

Topic

- CGI is the original, simple embedded web server scripting
- Language designers
 - Compiler design is one of the toughest programming challenges
 - Designing, implementing scripting languages is a smaller, perhaps more manageable subset
 - Good way for proto-language designers to cut their teeth
 - For more sophisticated scripting languages, the gap is probably a lot less
 - Not saying that Guido von Rossum, others, are not as capable as Stroustrup, peers
 - Just saying that there is a potential shallow end for someone just getting started
- Overall, hackers working with scripting languages can get further and further with them
 - Used to be that the limitations of scripting languages meant you had to move quickly on to system languages
 - With standalone scripting languages, less need to do so
 - With popularity of web applications, whole class of programs that are fine implemented as scripting languages
 - Still need to be aware of limitations

39:22

• **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