Topic: Software Tools and the Basic Anatomy of C Programs

Approach: Dissect actual C programs

Overview and Intro:
Review the big picture: HTML, connector, script, tools
Demo: trainsched.html→trainsched.cgi→trainsched→tools
Need more tools to make it better

Crucial Concept: The Software Tool: Components and Connections
The Unix Philosophy
basic sequence: input, store, process, output
customization: activity can be directed by command line arguments

Facts about C Programs:
Most Unix tools are written in C
C is a compiled language
Program is written in a text editor (e.g. simple.c)
The *source code* is compiled into executable code
The executable code is run

Three Tools for the sched datafile
semi2tab1.c filter to change semicolons to tabs
capitalize1.c filter to upper case initials
We can now use them in pipelines and in our trainsched script
count_semis1.c analyze dataset : useful for cleaning

Looking at C Programs: The hello series
Computers do four things: input, storage, processing, output
Structure: Storage has structure, Code has structure
functions a C program consists of one or more functions
a function has: type, name, args, storage, code
storage simple variables: int, char, float (easy conversion)
varyations: long, short, double, unsigned
no Boolean: instead 0=false, non-0=non-false
arrays: sequence of storage boxes: defining, using
operations arithmetic(+,-,*,/), comparison, boolean
control flow while, for, if
functions defining, using, passing data, declaring
arguments args to functions: simple passed by value, arrays by ref
input/output numbers, characters, strings
strings an array of chars terminated with a null char

More trainsched Processing Tools:
count_semis2.c a more informative data analyzer
empties.c find fields with no data
missing.c find lines missing one or more tags
outoforder.c find lines with tags in wrong order

Good Programming Tip #1:
Use *gcc -Wall* to compile and fix each warning

Copyright (c) 2017 Bruce Molay