A tour of my junkcode directory

Andrew Tridgell
tridge@samba.org

"the best example code is your own"

What is junkcode?

- “junkcode” is little snippets of code written to test an idea or just to see how something works.
- Most programmers write bits of junkcode at one time or another. It's an important part of learning to be a better programmer.
- In this talk I hope to convince you to value your junkcode, and to keep it rather than deleting it when it has served its initial purpose. Today's junk can help you build tomorrows killer app.

1993 - my first junkcode

```
#include <stdio.h>
#include <sys/time.h>

main()
{
struct timeval tp1;
struct timeval tp2;
int i;
for (i=0;i<10;i++)
  {
    gettimeofday(&tp1,NULL);
    usleep(1*i);
    gettimeofday(&tp2,NULL);

    printf("diff=%d\n",
           (tp2.tv_sec-tp1.tv_sec)*1000000 +
          (tp2.tv_usec-tp1.tv_usec));
  }
}
```

Reasons for writing junkcode

- There are many reasons for writing a snippet of junkcode:
    - testing how a library function works
    - testing your understanding of a language feature
    - seeing if you can trigger a bug
    - exploring an idea for a larger program
    - buggering around, trying to avoid real work

Reasons for keeping junkcode

- People tend to forget stuff, and I forget stuff really fast.
- When I go to use a function that I've used before but have forgotten how to use, I grep in my junkcode directory.
- I find that seeing my own code doing something tends to remind me faster than looking at someone elses code

junkcode as a toolbox

- Your junkcode directory forms a type of toolbox.
- I find that I often need some little linux utility program to solve a problem, and usually find that I can build the utility quickly by reusing exiting junkcode
- Instead of telling people “I once wrote a program that does that” I can say “here is a program that does that”. A great way to impress the boss!

on the left of the bus ....

- Welcome to the junkcode bus! Of over 400 progs I've selected these for a closer look:

• segv_handler	- a system wide segv handler
• mail.runner	- send/recv email with bash/rsync/ssh
• i	- the smallest app accepted into debian?
• tserver	- server side scripting for bash junkies
• 3dttt	- fun with OpenGL !
• gz_extract	- find embedded bits of gzip data
• socklib	- a simple network benchmark
• genstruct	- like Data::Dumper for C
• tphdisk	- a simple hibernation setup program

segv_handler

```
static int segv_handler(int sig)
{
        char cmd[100];
        char progname[100];
        char *p;
        int n;
        n = readlink("/proc/self/exe",progname,sizeof(progname));
        progname[n] = 0;
        p = strrchr(progname, '/');
        snprintf(cmd, sizeof(cmd),
                 "backtrace %d > /tmp/segv_%s.%d.out 2>&1",
                 (int)getpid(), p+1, (int)getpid());
        system(cmd);
        _exit(1);
}
void _init(void)
{
        signal(SIGSEGV, segv_handler);
}
```

mail.runner

- Started as a quick hack to to send/receive email over rsync/ssh.
- It's still a quick hack, but now at least a fairly good quality quick hack.
    - completely removes the need for a local MTA
    - very efficient over high latency links
    - only link to the world needed is ssh
    - can continue partial transfers efficiently

iprint

```
static void print_one(unsigned long long v)
{
   printf("%llu 0x%llX 0%llo", v, v, v);
   if (v < 256 && isprint(v)) printf(" '%c'", (unsigned char)v);
   printf("\n");
}

int main(int argc,char *argv[])
{
   int i;
   for (i=1;i<argc;i++) {
      unsigned char *p;
      unsigned long long v = strtoull(argv[i], (char **)&p, 0);
      if (p == (unsigned char *)argv[i]) for (; *p ; p++)
         print_one(*p);
      else print_one(v);
   }
   return 0;
}
```

tserver

- tserver is a tiny webserver, meant for embedded applications
    - includes a server side scripting language based on bash
    - can also be used via CGI with an existing web server
    - supports recursive evaluation
- http://junkcode.samba.org/ is an example of tserver in action

3dttt

- 3dttt is a OpenGL 3D Tic-Tac-Toe program
- Written to teach me about OpenGL and game programming
- 3dttt was the program that got me into game programming, leading to KnightCap
- I have reused code from 3dttt several times for various 3D applications

gz_extract

- my most recent junkcode!
- searches a file for gzip signatures, and dumps all candidate gzip data as separate files
- useful for extracting gzip data (such as initrd images) from firmware files

genstruct

- A system for marshalling/unmarshalling arbitrary data structures in C
  - uses a perl script to pre-process C headers into a C table
  - table used at runtime to marshall/unmarshall C structures
  - uses a human readable marshalling format
  - robust to structure shape changes
- genstruct forms the basis for 'lasker', my internet chess server. Also used in Samba3

tphdisk

- A system for setting up laptop hibernation files and partitions
- no need to boot to DOS
- very simple and safe operation

junkcode graduates

- Sometimes junkcode has escaped!
    - tdb - the trivial database
    - dbench - useful benchmark emulation
    - ccache - a compiler cache
    - pidl - an IDL compiler and code generator
    - rzip - a compression program for large files

lots more ....

- Have a look at http://junkcode.samba.org/
- Start your own junkcode area! You won't regret it.

slides at http://samba.org/ftp/tridge/talks/junkcode.pdf