

The Walnut Kernel

A SummerVac 2003 Project
Stewart Smith

Walnut Kernel History

- Original Password Capability System
 - M. Anderson, R. D. Pose, and C S. Wallace
 - The hardware no longer exists :
 - Access control up to word granularity
- Maurice Castro created the Walnut Kernel for his PHD with much help from others
 - Password Capability based
 - Available on standard PC Hardware*
 - Access control up to page granularity
- Has not been touched much since 1996

* Meaning a 486 with very specific disk configuration

The Basic Idea

- Large 128 bit address space
- Objects are mapped into processes address space
- Parts of password capability
 - Volume ID
 - Object ID
 - Password 1
 - Password 2
- Can derive capabilities for equal or lesser rights
- All objects are persistent
- Processes are Objects
- This mean processes are persistent

What We Found

- A MESS!!!!
- Directories full of code on both sutekh and bruce
- Which was current? What worked?
- Walnut Testbed booted, and you could do some things
 - But where was it compiled from?
 - How do we get GLui running?
 - Result:
 - Lot's of guesswork on the interface to 'drive'

```
% ls
Makefile
articles
cvsroot
doc
klibc-functions.txt
klibc-functions.txt~
klibc.c
klibc.c~
klibc.h
klibc.h~
klibc.o
mahoney.html
mkbimage.txt
portability_issues.txt
```

```
portability_issues.txt~
presentation.cwk
stdalone-osx-build.patch
sutekh
sutekh RAM configuration table.html
sutekh RAM configuration table.html~
sutekh RAM configuration table.ooutline
sutekh RAM configuration.graffle
sutekh internals.graffle
sutekh.jpg
sutekhSW1.png
walnut
www.linuxgazette.com
```

```
% ls  
%
```

On the weekend, I reconstructed a lot

(and started the presentation)

(and started a report)

(and went away to Inverloch - **AWAY FROM INTERNET**)



- After <4hrs sleep/night
- 1/2 slab of Becks
- killing 2 machines
- running out of disk space
- cursing GCC 2.96

Walnut is back

Build System

- Sutekh was running FreeBSD 1.1
 - Dual sets of makefiles for DOS and BSD
 - Incomplete Makefiles some parts of the system didn't build from a 'make f makefile.bsd'

What I Fixed

- 'make f makefile.bsd' now builds a Walnut Kernel
- You can now compile and link Walnut on newer BSD systems and Linux
- You now get an ELF executable and the possibility of Multiboot compliance

Really Nasty Things

- Custom boot sector
- Nasty UI drive
- Nasty code drive.c
- Links against libc
- Doesn't like use more than 8MB Memory
- Doesn't like 'large' drives >256MB
 - Stanley hacked it onto a 500MB drive
 - It boots once, then overwrites the partition table

What I Fixed

- No longer links against libc
- klibc by Stewart Smith
 - Unlike GNU libc, it's not K&R C :

Code Stats

- ~40,000 LOC
- ~2,460 goto's
- ~3270 comments

```

#define MAXKEY 90
Sw lastkey;      /* 1+index of last defined keyword */
Sw keynum[128] = {0,0};
Sw keyindex [128];
char *key[MAXKEY] =
    {
        "vol ", "ser ", "p1  ", "p2  ", "sr  ",
        "ur   ", "bs  ", "lim ", "mon ", "typ ",
        "moff", "msz ", "mcp ", "ofst", "subp",
        "cidx", "when",
        "newo", "newc", "fobj", "fcap", "dcap",
        "dder", "pobj", "save", "shrk", "nudg",
        "read", "writ", "wait", "file", "ldcp",
        "rdlw", "wrlw", "nprc", "cpid", "unld",
        "send", "recv", "exrd", "exwr", "exsd",
        "bank", "paot", "pvol", "rstr", "wrst",
        "exec", "copy", "push", "diag", "bfld",
        "afex", "fail", "shut", "renm", "rdst",
        "nsub", "dsub", "load", "osnd", "thot",
        "acpt", "clmb", "cobj", ":",  ", "jump",
        "mov ", "add ", "mul ", "div ", "and ",
        "or  ", "shft", "nono", "skeq", "skne",
        "skge", "skgt", "skle", "sklt", "movc",
        " ", ". ", "; ", "peek", "pmix",
    };

Sw lastpar = 17; /* Number of 'key' words showing a value */

```


So what do you do?

Give Up!

Some of Walnut is well documented

Some of it isn't.

```
reading:          /* We must be waiting for completion of a disc read */
                  /* When read complete, last 2 bits of page state = 3 */
                  if (fizent1->state != 0x13) {scr->error = -1504; goto quitlock;}
/*E01504-         Fixdope: disc read of dope unfinished */
                  fizent1->state = 0x10;
                  thead->stage = 2;
```

```
readdone:
/*              There is a page of dope in the buffer. If it is the first part of
                the dope, we should check that it is the right object, and if so
                make a new dope block and start copying into it.  */
/*              Set pointer to buffer */
                wpl = (Sw *) framenumadd (sw1) ; /* Ptr to buffer */
                if (head) goto havearea;
/*              The first part of the dope is in the buffer.*/
                bhead = (Header *) wpl;
/*              Check it is dope for right object          */
                if (bhead->magic != DOPEBOTMAGIC) {scr->error = 1505;
/*E01505         'Dope' block from disc proves not to be dope  */
                goto badblock; }
```

Portability

- Originally Designed to be portable across different architectures.
- Idea was for summervac to be porting Walnut to PowerPC
 - Obviously, this didn't happen.

Word Size

- All words are not created equal!
- 16bits? 32bits? 64bits?
- Rough estimate is that there are over 1000 LOC that explicitly mention 'word' and not '32bits' or similar .
- There may be other lines which assume 32bit word size
- We're in trouble if we ever want to go to 64bit

Page Size

- All pages are not created equal!
- x86 and most other modern architectures use 4kb pages
- Older machines had much smaller page sizes 512byte
- In the future, it is safe to assume that we will be dealing with larger pages
- We have access control on page boundaries.
- Security and programs may break if we use volumes with different page sizes

Rent Collector

- This is one thing that now works
- Stanley and Jan got this going
- Not a lot of code
- Steps through each object on a volume and charges rent based on size and time since last rent collected
- Many problems with how to track when rent was collected
- 'Unused' fields in Kernel Data structures

The Future...

- 'Advanced' Hardware support
 - Disks
- and other things i'll talk about 'cause i can't finish my notes now.....