Project 1: Getting Acquainted

Andrew Chase

OS II

Spring 2015

Abstract: Describes work done to build and run a virtual kernel, as well as develop and test a simple concurrency demo.

CONTENTS

I. Work Log

Date	Work Done
Apr 7	Build Kernel and run on os-class
Apr 8	Write Concurrency solution
April 11	Write this paper

II. Kernel Setup

A. Command List

The following was the bash history for setting up the Kernel.

```
cd /scratch/spring2016/
ls
mkdir cs444-017
ls
cd cs444-017
ls
git init
git clone git://git.yoctoproject.org/linux-yocto-3.14
git clone --depth 1 git://git.yoctoproject.org/linux-yocto-3.14 v3.14.26
ls
cd v3.14.26/
ls
source /scratch/opt/environment-setup-i586-poky-linux
less /scratch/opt/environment-setup-i586-poky-linux
make
make -j4 all
cp -r /scratch/spring2015/files/config-3.14.26-yocto-qemu .config
cd .config
ls
man cp
cp -r /scratch/spring2015/files/config-3.14.26-yocto-qemu .config/
cp -r /scratch/spring2015/files/config-3.14.26-yocto-qemu .config
less .config
make -j4 all
make clean
```

```
make -j4 all
exit
ssh chasean@os-class.engr.oregonstate.edu
exit
ls
echo "cd /scratch/spring2016/cs444-017" > os
chmod a+x os
./os
ls
. os
ls
cd
ls
ls -a
vi .profile
vi .bash_profile
vi .bashrc
exit
vi .bashr
vi .bashrc
exit
ssh chasean@os-class.engr.oregonstate.edu
. os
ls
vi start-qemu
# qemu-system-i386 -gdb tcp::5517 -S -nographic -kernel arch/x86/boot/bzImage -drive fi
cp /scratch/spring2016/files/bzImage-qemux86.bin .
cp /scratch/spring2016/files/core-image-lsb-sdk-qemux86.ext3 .
ls
cd v3.14.26/
ls
cd kernel/
ls
cd ..
```

```
ls
less README
cd kernel/
ls
cd ..
ls
less Kbuild
less Makefile
cd ..
ls
tree
tree  > t
less t
cd v3.14.26/
ls
cd ..
ls
less t
rm t
mv start-qemu core-image-lsb-sdk-qemux86.ext3 bzImage-qemux86.bin v3.14.26/
ls
cd v3.14.26/
. start-qemu
source /scratch/opt/environment-setup-i586-poky-linux
. start-qemu
ls
. os
cd v3.14.26/
. /scratch/opt/environment-setup-i586-poky-linux
$GDB 5517
man gdb
ls
echo $GDB
vi $GDB
```

```
vi `echo $GDB`
which `echo $GDB`
/scratch/opt/sysroots/x86_64-pokysdk-linux/usr/bin/i586-poky-linux/i586-poky-linux-gdb
gdb 127.0.0.1:5517 vmlinux
ls
gdb 127.0.0.1:5517 bzImage-qemux86.bin
ls
gdb 127.0.0.1:5517
gdb 127.0.0.1:5517 core-image-lsb-sdk-qemux86.ext3
ls
$GDB 127.0.0.1:5517
$GDB 127.0.0.1:5517 bzImage-qemux86.bin
vi start-qemu
screen -R q
less
ps -u chasan
ps
ps -u chasean
$GDB 127.0.0.1:5517 2018
. /scratch/opt/environment-setup-i586-poky-linux
$GDB 127.0.0.1:5517 2018
$GDB :5517 2018
$GDB 2018
$GDB 127.0.0.1:5517
$GDB --help | less
ls
. os
ls
cd v3.14.26/
$GDB 127.0.0.1:5517 arch/x86/boot/bzImage
$GDB
ps
ps -u chasean
kill 2018
```

```
exit
. start-qemu
. /scratch/opt/environment-setup-i586-poky-linux
. start-qemu
vi start-qemu
. start-qemu
exit
screen -r q
exit
```

B. *Qemu command line flags*

Flag	**Meaning**
-gdb tcp::????	Wait for gdb connection on tcp:???? before continuing
-S	Freeze CPU execution on startup (use gdb to continue)
-nographic	QEMU has a graphical output system. This flag disables it
-kernel bzImage-qemux86.bin	Use selected bzimage as the system Kernel
-drive ...	Use selected drive and input file as the virtual hard drive for the machine
-enable-kvm	Enabled hardware assisted virtualization using special kernel-level optimizations
-net none	Disable networking
-usb	Enable usb driver support
-localtime	Sets virtual clock to the system clock; Deprecated 2009 in qemu commit 1ed2fc1
-no-reboot	When system shuts down, exit qemu instead of rebooting the virtual machine
-append ...	Kernels accept command line options, use these options specifed

III. Concurrency Exercise: Questions

A. *What do you think the main point of this assignment is?*

I think the main point of the assignment is to review writing, compiling, and testing low-level code which students will be doing in this course.

B. *How did you personally approach the problem? Design decisions, algorithm, etc.*

I used the low-level Pipe construct, after making sure that it was thread-safe for the purposes of the assignment. The use of pipes is simple and easy for programmers to understand. The trade-off for pipes was that it only works in a thread-safe way for small message sizes and is tied to a specific operating system level api, but the advantage of a pipe are that they are very easy to understand and implement.

C. How did you ensure your solution was correct? Testing details, for instance.

I ran the solution and inserted system print log statements to make sure the system was behaving as desired.

D. What did you learn?

That pipes can be used for concurrency, that newer Intel chips have randomization instructions, and a lot about the cmake build tool.

IV. Concurrency Exercise: File Tree

```
.
| - CMakeLists.txt
| - get_random.c
| - get_random.h
| - main.c
| - vendor
    | - drng
    | | - LICENSE
    | | - README
    | | - config.h
    | | - cpuid.c
    | | - cpuid.h
    | | - drng.c
    | | - drng.h
    | - mt19937ar.c
    | - mt19937ar.h
    | - not_using_drng
    | | - using_drng.h
    | - test_drng
    | | - test.c
    | - test_drng_bin
    | | - CMakeFiles
    |     | - CMakeTmp
    | - using_drng
        | - using_drng.h
```

```
8 directories, 16 files
```

V. Concurrency Exercise: Git Log

acronym	meaning
V	`version`
tag	`git tag`
MF	Number of `modified files`.
AL	Number of `added lines`.
DL	Number of `deleted lines`.

V	tag	date	commit message	MF	AL	DL
1		2016-04-07	init	2	15	0
2		2016-04-07	Add pthreads from pthreads example	3	42	11
3		2016-04-07	Stub consumers/producers functions and add pipe.	1	47	14
4		2016-04-07	Add random vendor files and interface	12	1112	2
5		2016-04-07	Implement randomness into producer	1	18	5
6		2016-04-08	Use Try_Compile to decide random at compile time	6	50	14

VI. Concurrency Exercise: Code Listings

A. *main.c*

```
/*


REFERENCES:
* https://computing.llnl.gov/tutorials/pthreads/samples/hello.c
* http://mij.oltrelinux.com/devel/unixprg/
* https://stackoverflow.com/questions/1620918/cmake-and-libpthread
* http://www.tutorialspoint.com/cprogramming/c_structures.htm
* https://stackoverflow.com/questions/12657962/how-do-i-generate-a-random-number-betwee
* https://stackoverflow.com/questions/4975340/int-to-unsigned-int-conversion


*/
```

```c
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include "get_random.h"

#define NUM_THREADS    2

struct WorkUnit {
    unsigned int unitNumber;
    unsigned int workTime;
};

int workPipe[2];

#define readPipe workPipe[0]
#define writePipe workPipe[1]

void *consumer(void *threadId) {
    struct WorkUnit work_message;
    long tid;
    ssize_t ret;
    tid = (long) threadId;

    for (; ;) {
        ret = read(readPipe, &work_message, sizeof(struct WorkUnit));
        if (ret == 0)
            break;
        sleep(work_message.workTime);
        printf("Thread #%ld consumed %u. Took %u seconds.\n", tid, work_message.unitNumb

    };
    pthread_exit(NULL);
}
```

```
#pragma clang diagnostic push
#pragma clang diagnostic ignored "-Wmissing-noreturn"

void producer() {
    struct WorkUnit producerMessage;
    unsigned int producerWaitTime;
    for (; ;) {
        producerMessage.unitNumber = get_random();
        producerMessage.workTime = get_random_between(2, 9);
        producerWaitTime = get_random_between(3, 7);
        write(writePipe, &producerMessage, sizeof(struct WorkUnit));
        sleep(producerWaitTime);
    }
}

#pragma clang diagnostic pop

int main(int argc, char *argv[]) {
    pthread_t threads[NUM_THREADS];
    int rc;
    long t;

    pipe(workPipe);

    for (t = 0; t < NUM_THREADS; t++) {
        printf("In main: creating thread %ld\n", t);
        rc = pthread_create(&threads[t], NULL, consumer, (void *) t);
        if (rc) {
            printf("ERROR; return code from pthread_create() is %d\n", rc);
            exit(-1);
        }
    }
```

```
    producer();
    pthread_exit(NULL);
}
```

B. get_random.c

```
#include <stdio.h>
#include "using_drng.h"

#if USING_DRNG
#include "vendor/drng/drng.h"
#else

#include "vendor/mt19937ar.h"

#endif

unsigned int get_random() {
#if USING_DRNG
    // Use intel drandr
    uint32_t rand[1];
    if (rdrand_get_n_uints(1, rand) == 0) {
        fprintf(stderr, "Random values not available\n");
        return 1;
    }
    return (unsigned int) rand[0];
#else
    // Use twister
    return (unsigned int) genrand_int32();
#endif
}

unsigned int get_random_between(int min, int max) {
    return get_random() % (max - min + 1) + min;
}
```

C. mt19937ar.c

```
/*
   A C-program for MT19937, with initialization improved 2002/1/26.
   Coded by Takuji Nishimura and Makoto Matsumoto.

   Before using, initialize the state by using init_genrand(seed)
   or init_by_array(init_key, key_length).

   Copyright (C) 1997 - 2002, Makoto Matsumoto and Takuji Nishimura,
   All rights reserved.
   Copyright (C) 2005, Mutsuo Saito,
   All rights reserved.

   Redistribution and use in source and binary forms, with or without
   modification, are permitted provided that the following conditions
   are met:

     1. Redistributions of source code must retain the above copyright
        notice, this list of conditions and the following disclaimer.

     2. Redistributions in binary form must reproduce the above copyright
        notice, this list of conditions and the following disclaimer in the
        documentation and/or other materials provided with the distribution.

     3. The names of its contributors may not be used to endorse or promote
        products derived from this software without specific prior written
        permission.

   THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
   "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
   LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
   A PARTICULAR PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE COPYRIGHT OWNER OR
   CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
   EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
```

```
   PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
   PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
   LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
   NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
   SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.


   Any feedback is very welcome.
   http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/emt.html
   email: m-mat @ math.sci.hiroshima-u.ac.jp (remove space)
*/

#include <stdio.h>
#include "mt19937ar.h"

/* Period parameters */
#define N 624
#define M 397
#define MATRIX_A 0x9908b0dfUL   /* constant vector a */
#define UPPER_MASK 0x80000000UL /* most significant w-r bits */
#define LOWER_MASK 0x7fffffffUL /* least significant r bits */

static unsigned long mt[N]; /* the array for the state vector  */
static int mti=N+1; /* mti==N+1 means mt[N] is not initialized */

/* initializes mt[N] with a seed */
void init_genrand(unsigned long s)
{
    mt[0]= s & 0xffffffffUL;
    for (mti=1; mti<N; mti++) {
        mt[mti] =
            (1812433253UL * (mt[mti-1] ^ (mt[mti-1] >> 30)) + mti);
        /* See Knuth TAOCP Vol2. 3rd Ed. P.106 for multiplier. */
        /* In the previous versions, MSBs of the seed affect   */
```

```
        /* only MSBs of the array mt[].                          */
        /* 2002/01/09 modified by Makoto Matsumoto               */
        mt[mti] &= 0xffffffffUL;
        /* for >32 bit machines */
    }
}

/* initialize by an array with array-length */
/* init_key is the array for initializing keys */
/* key_length is its length */
/* slight change for C++, 2004/2/26 */
void init_by_array(unsigned long init_key[], int key_length)
{
    int i, j, k;
    init_genrand(19650218UL);
    i=1; j=0;
    k = (N>key_length ? N : key_length);
    for (; k; k--) {
        mt[i] = (mt[i] ^ ((mt[i-1] ^ (mt[i-1] >> 30)) * 1664525UL))
          + init_key[j] + j; /* non linear */
        mt[i] &= 0xffffffffUL; /* for WORDSIZE > 32 machines */
        i++; j++;
        if (i>=N) { mt[0] = mt[N-1]; i=1; }
        if (j>=key_length) j=0;
    }
    for (k=N-1; k; k--) {
        mt[i] = (mt[i] ^ ((mt[i-1] ^ (mt[i-1] >> 30)) * 1566083941UL))
          - i; /* non linear */
        mt[i] &= 0xffffffffUL; /* for WORDSIZE > 32 machines */
        i++;
        if (i>=N) { mt[0] = mt[N-1]; i=1; }
    }

    mt[0] = 0x80000000UL; /* MSB is 1; assuring non-zero initial array */
```

```
}

/* generates a random number on [0,0xffffffff]-interval */
unsigned long genrand_int32(void)
{
    unsigned long y;
    static unsigned long mag01[2]={0x0UL, MATRIX_A};
    /* mag01[x] = x * MATRIX_A  for x=0,1 */

    if (mti >= N) { /* generate N words at one time */
        int kk;

        if (mti == N+1)   /* if init_genrand() has not been called, */
            init_genrand(5489UL); /* a default initial seed is used */

        for (kk=0;kk<N-M;kk++) {
            y = (mt[kk]&UPPER_MASK)|(mt[kk+1]&LOWER_MASK);
            mt[kk] = mt[kk+M] ^ (y >> 1) ^ mag01[y & 0x1UL];
        }
        for (;kk<N-1;kk++) {
            y = (mt[kk]&UPPER_MASK)|(mt[kk+1]&LOWER_MASK);
            mt[kk] = mt[kk+(M-N)] ^ (y >> 1) ^ mag01[y & 0x1UL];
        }
        y = (mt[N-1]&UPPER_MASK)|(mt[0]&LOWER_MASK);
        mt[N-1] = mt[M-1] ^ (y >> 1) ^ mag01[y & 0x1UL];

        mti = 0;
    }

    y = mt[mti++];

    /* Tempering */
    y ^= (y >> 11);
    y ^= (y << 7) & 0x9d2c5680UL;
```

```
    y ^= (y << 15) & 0xefc60000UL;
    y ^= (y >> 18);

    return y;
}

/* generates a random number on [0,0x7fffffff]-interval */
long genrand_int31(void)
{
    return (long)(genrand_int32()>>1);
}

/* generates a random number on [0,1]-real-interval */
double genrand_real1(void)
{
    return genrand_int32()*(1.0/4294967295.0);
    /* divided by 2^32-1 */
}

/* generates a random number on [0,1)-real-interval */
double genrand_real2(void)
{
    return genrand_int32()*(1.0/4294967296.0);
    /* divided by 2^32 */
}

/* generates a random number on (0,1)-real-interval */
double genrand_real3(void)
{
    return (((double)genrand_int32()) + 0.5)*(1.0/4294967296.0);
    /* divided by 2^32 */
}

/* generates a random number on [0,1) with 53-bit resolution*/
```

```c
double genrand_res53(void)
{
    unsigned long a=genrand_int32()>>5, b=genrand_int32()>>6;
    return(a*67108864.0+b)*(1.0/9007199254740992.0);
}
/* These real versions are due to Isaku Wada, 2002/01/09 added */
```