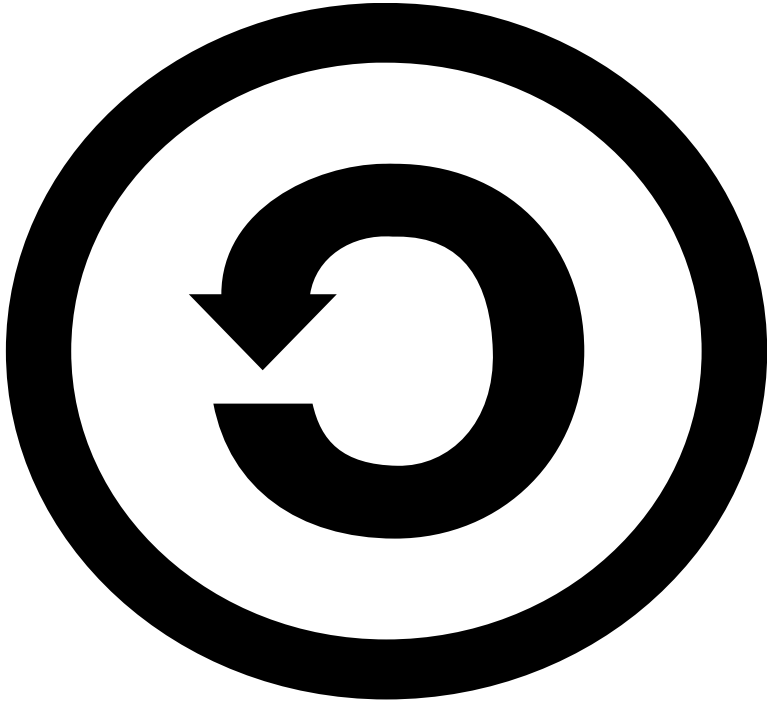
Kunsten af lave tilfældige tal



Agenda

1. Introduktion til emnet
2. Software
3. Hardware
4. Spørgsmål / Kommentarer / Diskussion

Agenda

1. **Introduktion til emnet**
2. Software
3. Hardware
4. Spørgsmål / Kommentarer / Diskussion

Hvorfor?

“For cryptography, game playing, sociological surveys and various scientic calculations, people often need series of numbers that are devoid of pattern.

And that is a tall order.

Generating true randomness is one of computer science’s most difficult challenges”

George Johnson i “New York Times”, 12. juni 2001

Hvor bruges tilfældige tal?

NEMID

Nøgle-kort f.eks. nemid med 148 forrykte koder på 6 cifre plus 4 cifret indeks-nummer.

Teknikken bag udvælgelsen af tallene er ukendt.

Hvor bruges tilfældige tal?

En-armet tyveknægte etc.

Ifølge BEK nr 1302 af 15/12/2011

- § 4. Spillets udfald skal i hvert spil bero på **tilfældighed**.
- § 6. Spillets afvikling må kun finde sted ved **manuel påvirkning** af selve spilleautomaten.
  Dog tillades påvirkninger fra spilleren som hold, nudge og lignende, som **ikke indebærer færdighed**, men som beror på tilfældighed.
- § 12. Spilleautomatens deklarerede **tilbagebetalingsprocent**, der skal kunne aflæses på maskinen, må ikke være under **74 procent**.

Hvor bruges tilfældige tal?

Klasselotteriet

- Oprettet i 1753 af Frederik 5.
- Skulle drives af Det Kongelige Opfostringshus, som fik (det meste af) overskuddet.
- Staten manglende penge og Struense fik det overført til staten i 1771

Hvor bruges tilfældige tal?

Klasselotteriet

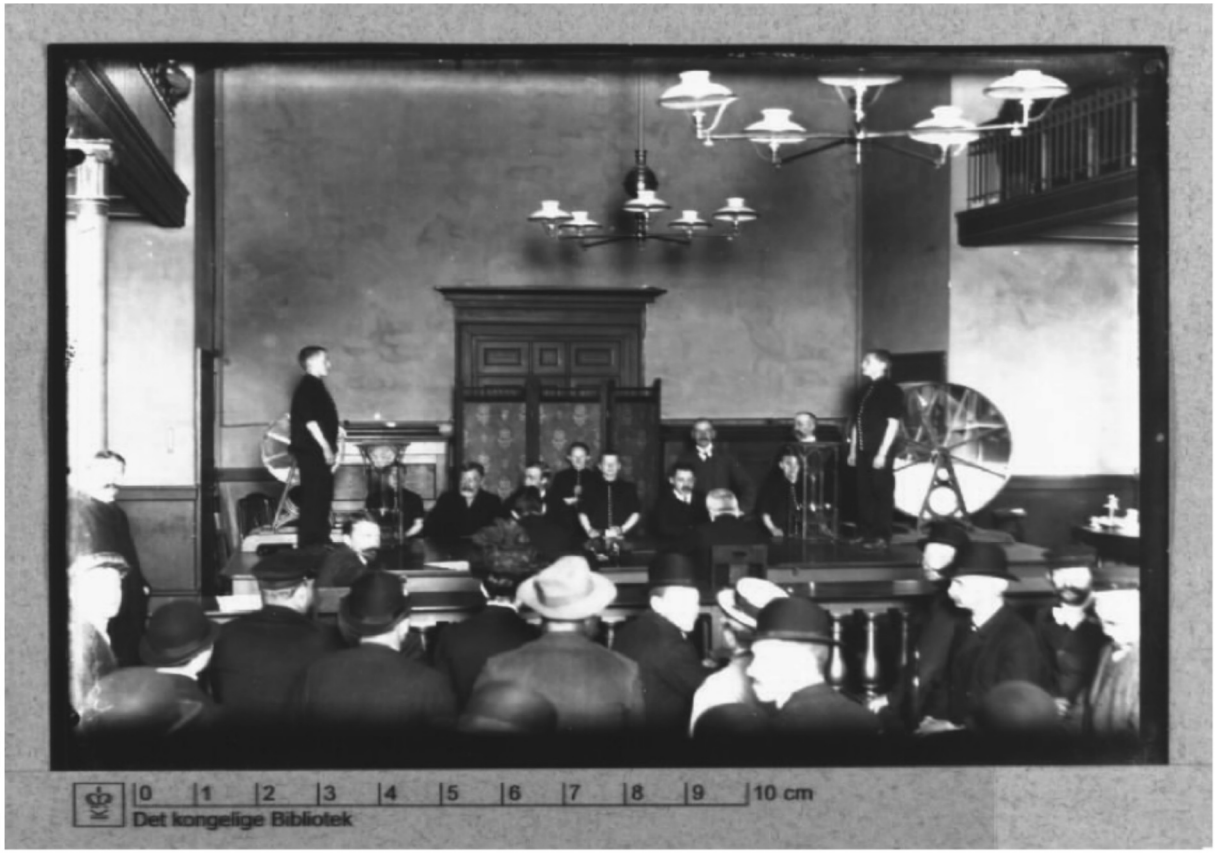
- Omsætter for omkring 760 millioner om året med betalingsprocent på 65%.
- 460.000 lodder, som kan deles ned til 1/8 dele.
- 40.000 lodder vinder hver måned.
- Rigtig mange små gevinster.

Hvor bruges tilfældige tal?

Klasselotteriet

- Daglige trækninger 15.-19. hver måned.
- Indtil 1995 manuel håndrulning af alle lodder og manual trækning

Hvor bruges tilfældige tal?



Hvor bruges tilfældige tal?

Klasselotteriet

- Fuld elektronisk trækning fra april 2000
- Til hver trækning bruges 48 tilfældige cifre fra Notarius Publicus, som Spillemyndigheden, medbringer i en forseglet kuvert.
- De 48 cifre tastes ind i trækningsprogrammet og herefter tager tilfældighedsgeneratoren over og parrer lodnumre og gevinster.

Hvor bruges tilfældige tal?

- ssh forbindelser
- https forbindelser
- bitcoin adresser
- machine learning algoritmer
- computerspil
- computational biology
- ...

Tilfældighed / determinisme

Findes helt tilfældige tal/begivenheder?

NEJ, hvis man mener at verden er 100% deterministisk.

Dvs. at alle fremtidige hændelser er fastlagt af tidligere hændelser.

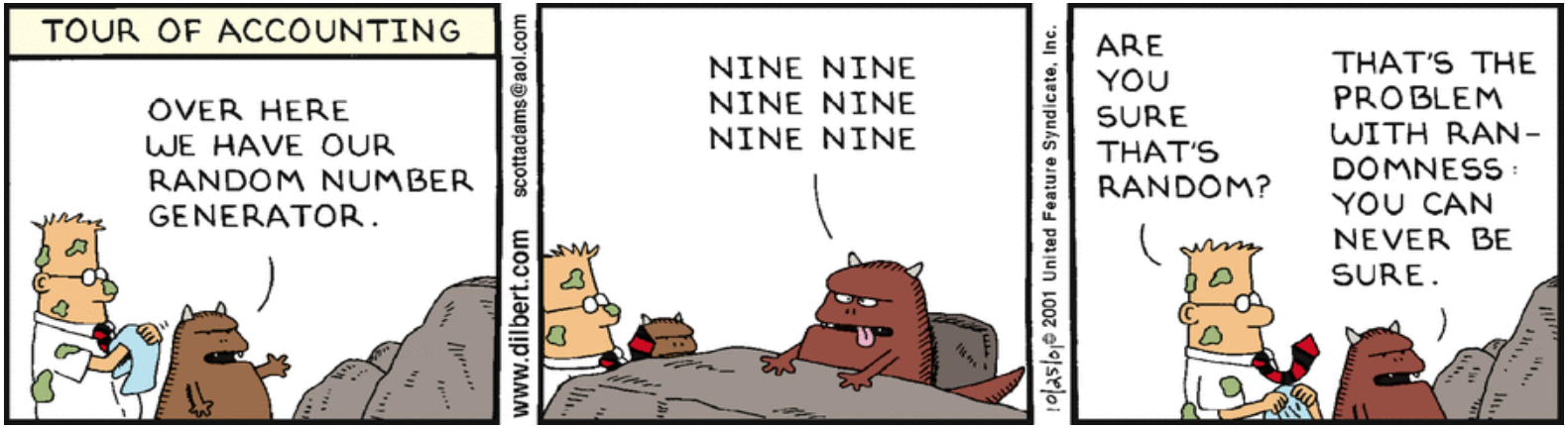
JA, hvis kvantemekanikken er en korrekt beskrivelse af stofs egenskaber ved atomære og sub-atomære nivauer.

Test for tilfældige

Der findes dårlige software og hardware random number generators (RNG).

Hvordan tester man RNG'er?

Svar: Statistiske analyser.



Test for tilfældighed

Bedste software til test er p.t. *dieharder* og *ent*

```
$ ent onerng.bin
Entropy = 8.000000 bits per byte.

Optimum compression would reduce the size
of this 1912602624 byte file by 0 percent.

Chi square distribution for 1912602624 samples is 288.66, and randomly
would exceed this value 7.23 percent of the times.

Arithmetic mean value of data bytes is 127.5038 (127.5 = random).
Monte Carlo value for Pi is 3.141474937 (error 0.00 percent).
Serial correlation coefficient is 0.000027 (totally uncorrelated = 0.0).
```

Test for tilfældighed

Bedste software til test er p.t. *dieharder* og *ent*

```
$ dieharder -a -k1 -s1 -g 201 -f onerng.bin

#==================================================================#
#            dieharder version 3.31.1 Copyright 2003 Robert G. Brown          #
#==================================================================#
   rng_name    |           filename             |rands/second|
 file_input_raw|                        onerng.bin|  1.38e+07  |
#==================================================================#
        test_name   |ntup| tsamples |psamples|  p-value |Assessment
#==================================================================#
   diehard_birthdays|   0|       100|     100|0.30764979|  PASSED
      diehard_operm5|   0|   1000000|     100|0.56619244|  PASSED
  diehard_rank_32x32|   0|     40000|     100|0.85658345|  PASSED
    diehard_rank_6x8|   0|    100000|     100|0.80921401|  PASSED
   diehard_bitstream|   0|   2097152|     100|0.50459144|  PASSED
        diehard_opso|   0|   2097152|     100|0.83276832|  PASSED
        diehard_oqso|   0|   2097152|     100|0.10426592|  PASSED
[...]
          sts_serial|  11|    100000|     100|0.00346564|   WEAK
[...]
      rgb_lagged_sum|  12|   1000000|     100|0.72187148|  PASSED
```

Agenda

1. Introduktion til emnet
2. **Software**
3. Hardware
4. Spørgsmål / Kommentarer / Diskussion

Software

Hvad er der galt med koden?

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
int main() {
    srand(time(NULL));
    for (int i = 0; i < 16; ++i) {
        printf("%d ", rand() % 100);
    }
    printf("\n");
}
```

Software

time(NULL) giver antal sekunder siden unix epoch.

- Forudsigeligt
- Lav opløsning

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
int main() {
    srand(time(NULL));
    for (int i = 0; i < 16; ++i) {
        printf("%d ", rand() % 100);
    }
    printf("\n");
}
```

Software

Tal fra ældre versioner *rand()* er ikke så tilfældige på de lavest bit.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
int main() {
    srand(time(NULL));
    for (int i = 0; i < 16; ++i) {
        printf("%d ", rand() % 100);
    }
    printf("\n");
}
```

Software

rand() giver tal i intervallet [0, 2^{32}-1].

100 går ikke op i 2^{32}

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
int main() {
    srand(time(NULL));
    for (int i = 0; i < 16; ++i) {
        printf("%d ", rand() % 100);
    }
    printf("\n");
}
```

Software

Forslag fra stackoverflow.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
int main() {
    srand(time(NULL));
    for (int i = 0; i < 16; ++i) {
        int src = rand();
        int dst = static_cast<int>((src * 1.0 / RAND_MAX) * 99);
        printf("%d ", dst);
    }
    printf("\n");
}
```

Software

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
int main() {
    srand(time(NULL));
    for (int i = 0; i < 16; ++i) {
        int src = rand();
        int dst = static_cast<int>((src * 1.0 / RAND_MAX) * 99);
        printf("%d ", dst);
    }
    printf("\n");
}
```

```
static_cast<int>((RAND_MAX-2) * 1.0 / RAND_MAX) * 99) → 98
static_cast<int>((RAND_MAX-1) * 1.0 / RAND_MAX) * 99) → 98
static_cast<int>((RAND_MAX  ) * 1.0 / RAND_MAX) * 99) → 99
```

Software

- /dev/random samler støj fra device drivers og andre kilder.
  - Indeholder ikke nødvendigvis data
  - Kan bruges som seed
  - Muligt at skrive ekstra data til filen
- /dev/urandom
  - Returnerer data fra en RNG baseret på /dev/random
  - Indeholder altid data
- CryptoAPI
  - Microsoft

Software

- haveged
  (HArdware Volatile Entropy Gathering and Expansion)
  - dæmon der putter data i /dev/random

Software

LCG (Linear congruential generator)

- Returnerer altid en ny værdi, der kun er baseret seneste værdi
- Nem at implementere
- Vigtigt at vælge godt værdier
- De laveste bit-værdier er ikke så tilfældige som de højeste bit-værdier

$$X_o = seed$$
$$X_{n+1} = (a \cdot X_n + c) \bmod m$$

Software

Lehmer random number generator:

- En version af LCG, hvor *c* er 0
  dvs. $X_{n+1} = (a \cdot X_n + c) \bmod m$
- Krav: *m* er p^k, hvor *p* er et primtal
- Krav: *a* er en primitiv rod modulus *n*

- ZX81 (1981), ZX Spectrum (1982-) brugte
  $a = 75$ og m = 65537
- IBM *randu* funktion brugte
  $a = 65539$ og m = 2^{31}
  ”System/360 Scientific Subroutine Package, Version III, Programmer's Manual. 1968”

Software

Pas på med at bruge indbygget PRNG!

Mange er rigtig dårlige.

- Perl - *rand()*
- Java - *java.util.Random* klassen
- Python før v2.3 - *random()*
- C - *rand()*, *drand48()*, *random()*
- Bogen “Numerical Recipes” *ran0()*, *ran1()*
- IBM - *RANDU*
  Meget brugt i starten af 1970’erne.
  “one of the most ill-conceived random number generators ever designed” - Donald E. Knuth
- ...

Software

Hvordan virker *java.util.Random?*

- Generere 48-bit tilfældige tal vha. en standard LCG:

```
private static final long multiplier = 0x5DEECE66DL;
private static final long addend = 0xBL;
private static final long mask = (1L << 48) - 1;
```

- De højeste 32 bit bliver returneret via *nextInt()*

```
nextseed = (oldseed * multiplier + addend) & mask;
return (int)(nextseed >>> 32);
```

- *nextlong()* returnerer en 64 bit-værdi baseret på to *nextInt()*

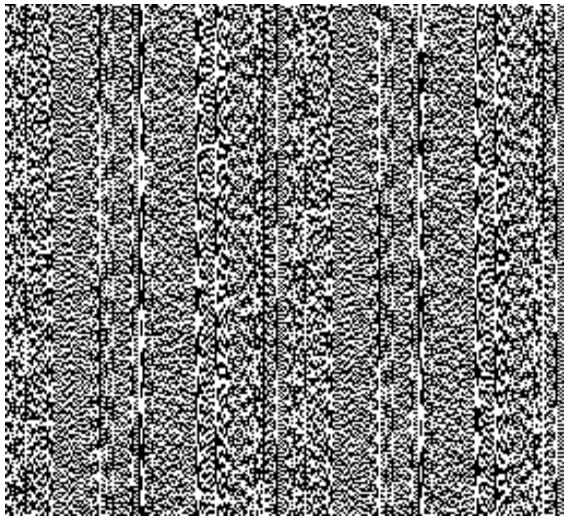
```
public long nextLong() {
  return ((long)(next()) << 32) + next();
}
```

Software

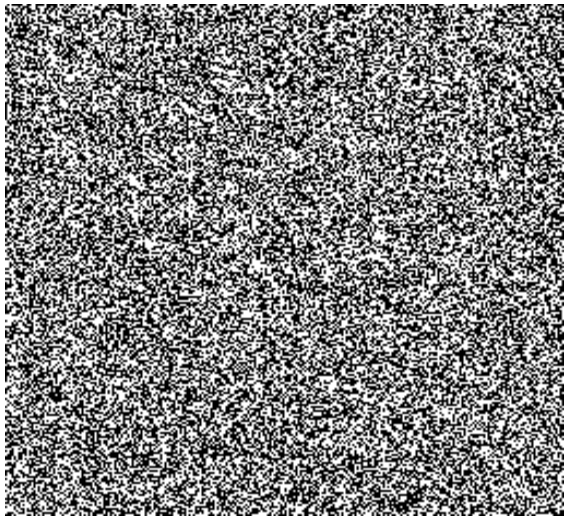
Hvad er der galt med *java.util.Random* ?

- Lav periode på kun 2^{48}
- *nextlong()* returnerer en 64 bit-værdi
  Men en del af de mulige 64 bit-værdier vil aldrig blive returneret med en 2^{48} periode.
- De laveste bit er ikke så tilfældige, som de højeste. Generelt et problem med LCG'er.

Lavest bit fra *nextInt()*



Højeste bit fra *nextInt()*

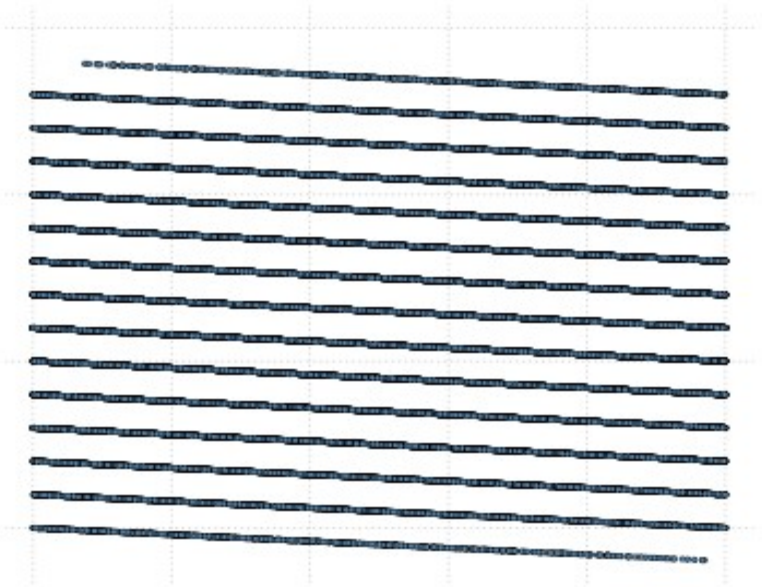


Software

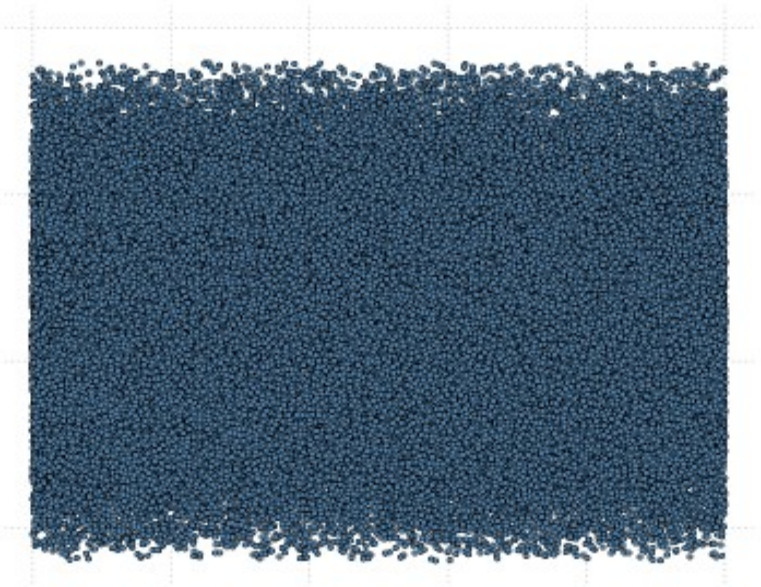
Hvad er der galt med *RANDU?*

- Lavest bit er alt 1 dvs. kun ulige tal.
- $X_{n+2} = 6 \cdot X_{n+1} - 9 \cdot X_n$
  Derfor vil en 3D afbildning af (X_n, X_{n+1}, X_{n+2}) lægge alle punkter i bare 15 planer.

RANDU



Standard rng



Software

Gode PRNG:

- “Mersenne Twister” *MT19937* og *MT19937-64* af Matsumoto og Nishimura (1997)
- KISS, Super KISS, MWC256, CMWC4096 (og mange flere) af George Marsaglia (1998-)
- Gruppen af “WELL” RNG af Panneton, L’Ecuyer, Matsumoto (2006)

Software

Mersenne Twister MT19937

- Standard RNG i nyere version af bl.a.:
  Excel, GLib, GMP, GNU Octave, MATLAB, PHP, Python, R
- Baseret på Mersenne primtallet $2^{19937}-1$
- Periode på $2^{19937}-1$
- Bruger et state-array på 624 32-bit unsigned integers
- Meget hurtig

Software

Super KISS

- Periode på hele $54767 \cdot 2^{1337279}$
- Returnerer 32-bit værdier
- Bruger et state-array på 41790 32-bit unsigned integers.

```
static unsigned int Q[41790];
static unsigned int indx = 41790, carry = 362436, xcng = 1236789, xs = 521288629;

unsigned int refill() {
  for (int i = 0; i<41790; i++) {
    unsigned long long t = 7010176ULL * Q[i] + carry;
    carry = (t>>32);
    Q[i] = -t;
  }
  indx = 1;
  return Q[0];
}
unsigned int SuperKISS() {
  xcng = 69069 * xcng + 123;
  xs ^= xs << 13;
  xs ^= xs >> 17;
  xs ^= xs >>  5;
  return (indx<41790 ? Q[indx++] : refill()) + xcng + xs;
}
```

Software

WELL (Well Equidistributed Long-period Linear)

- En familie af PRNG'er
- Overtager måske MT19937 plads som standard på et tidspunkt
- State array i størrelsen 512 til 44497 bit.

```
static unsigned long state[16]; /* initialize state to random bits */
static unsigned int index = 0;


/* return 32 bit random number */
unsigned long WELLRNG512(void) {
unsigned long a, b, c, d;
  a = state[index];
  c = state[(index+13)&15];
  b = a^c^(a<<16)^(c<<15);
  c = state[(index+9)&15];
  c ^= (c>>11);
  a = state[index] = b^c;
  d = a^((a<<5)&0xDA442D24UL);
  index = (index + 15)&15;
  a = state[index];
  state[index] = a^b^d^(a<<2)^(b<<18)^(c<<28);
  return state[index];
}
```

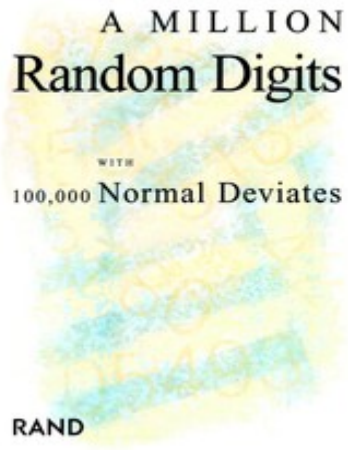
Agenda

1. Introduktion til emnet
2. Software
3. **Hardware**
4. Spørgsmål / Kommentarer / Diskussion

Hardware

“A MILLION Random Digits with 100,00 Normal Deviates”

- Forskning af RAND Corporation i 1940’erne og 1950’erne
- Skulle bruges til analyser bestilt af USA’s luftvåben
- Udkommet som bog i 1955, genudgivet i 2001
- Kunne også købes som 20,000 hulkort hvert med 50 cifre
- Første udgaven koster mange tusind kroner
- Kan downloades gratis og lovligt fra RAND



Hardware

“A MILLION Random Digits with 100,00 Normal Deviates”

- En pulsgenerator med omkring 100,000 puls per sekund
- Fem binære tæller udfra pulsgeneratoren
- Et cifre i sekunder
- En masse test-kørsler og problemer
- Fra maj til juni 1947 blev de endelige cifrene fremstillet

Hardware

“Erlang S - Statistiske tabeller” ved Allan C. Malmberg

- Dansk bog med bl.a. tilfældige tal
- Bliver stadigvæk udgivet
- Ikke så mange tilfældige tal
- Bedre indeksering af tallene

Hardware

Med hardware RNG’er kan det være nødvendigt at foretage en såkaldt “**Software Whitening**”

F.eks. hvis det ser ud til at der kommer flere 0 end 1 bit ud fra hardwaren eller omvendt.

Hardware

Software Whitening

En løsning foreslået af John von Neumann:

- Tag to bit af gangen
- Hvis de er “00” eller “11”, så smid dem væk
- Hvis de er “01”, så output “1”
- Hvis de er “10”, så output “0”

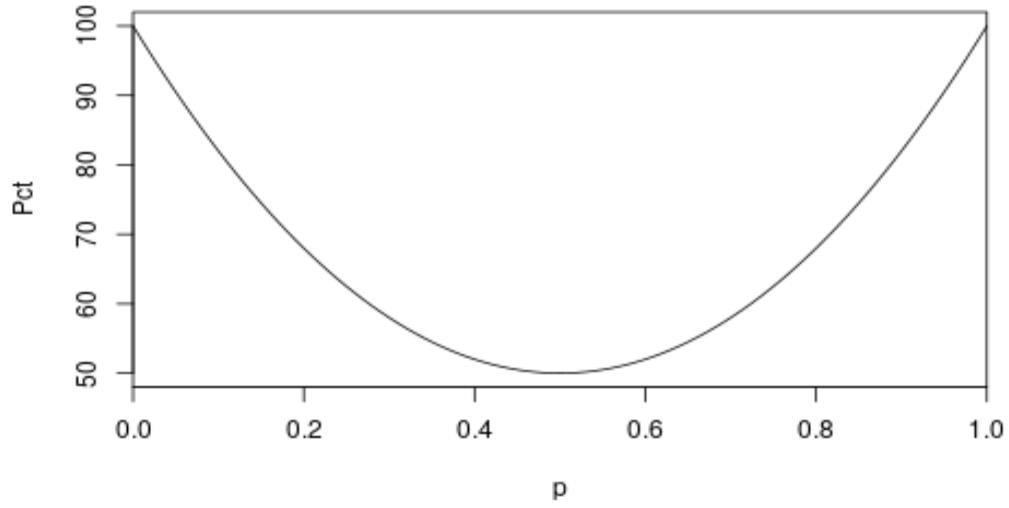
Hardware

Software Whitening

Analyse: Hvis 0 forekommer med p sandsynlighed, så forekommer 1 med $(1-p)$ sandsynlighed.

"01" forekommer med $p \cdot (1-p) = p - p^2$
"10" forekommer med $(1-p) \cdot p = p - p^2$

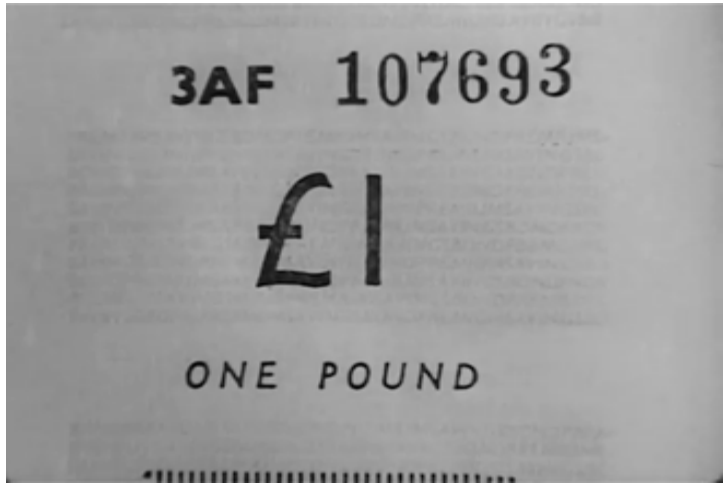
"00" forekommer med $p \cdot p = p^2$
"11" forekommer med $(1-p) \cdot (1-p) = 1 - 2 \cdot p + p^2$



Hardware

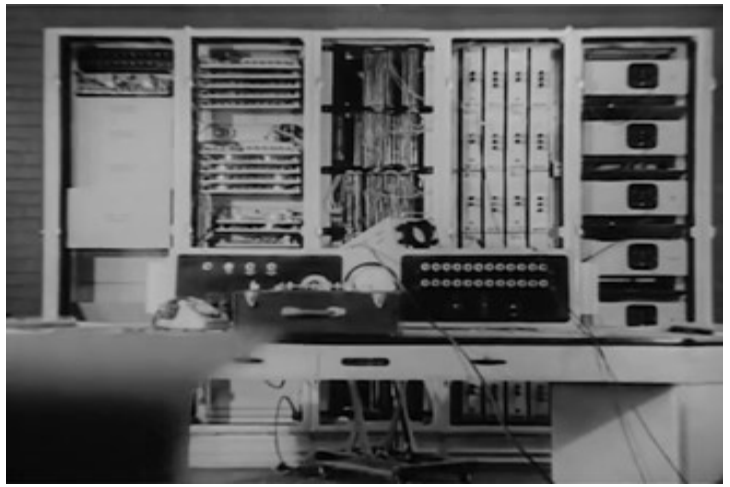
E.R.N.I.E. (“Electronic Random Number Indicator Equipment”)

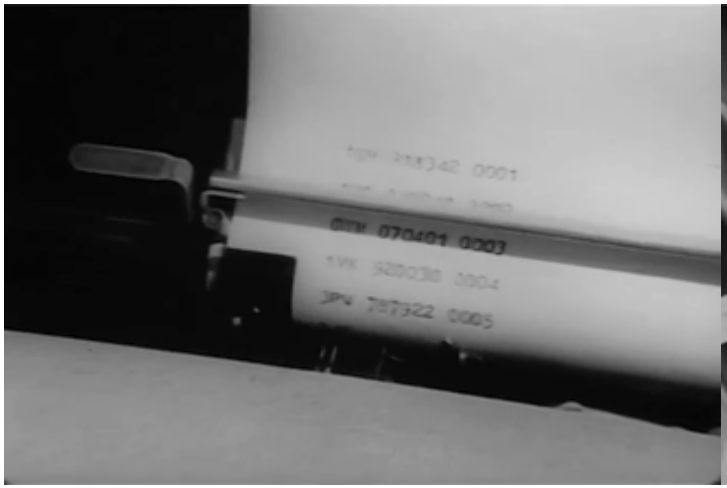
- Maskine som udtrækker vinderne af de engelske statsobligationer
- Gevinsten er renterne fra obligationerne
- Version 1 (1957-1972)
  - Brugte støj fra gas neonrør
  - Kunne genere 2000 vindere på en time
- Version 4 (2004-)
  - Bruger “thermal noise” fra en transistor



Hardware

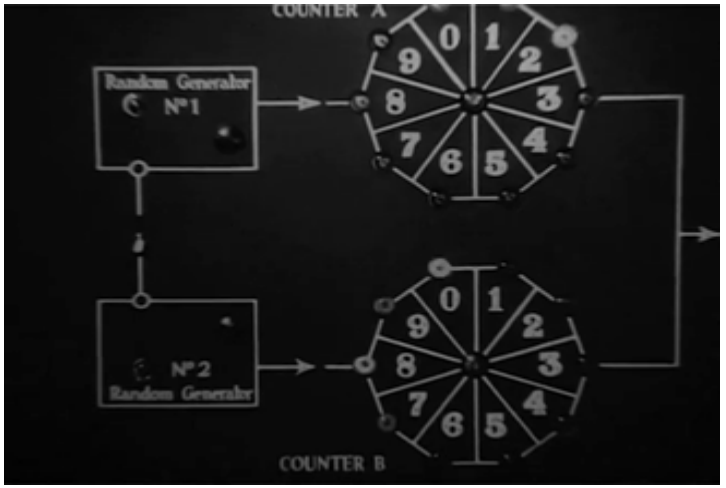
(Fra youtube: “The Importance of Being E. R. N. I. E.”)





Hardware

(Fra youtube: “The Importance of Being E. R. N. I. E.”)



Hardware

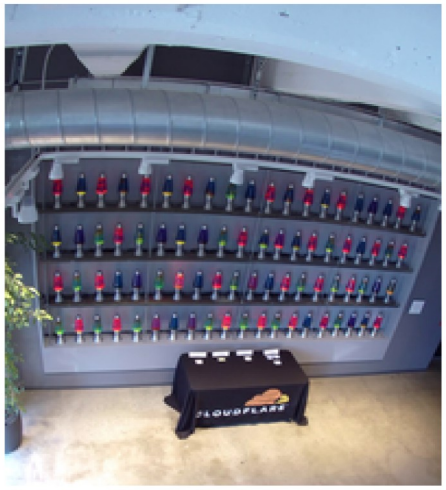
Lavarand

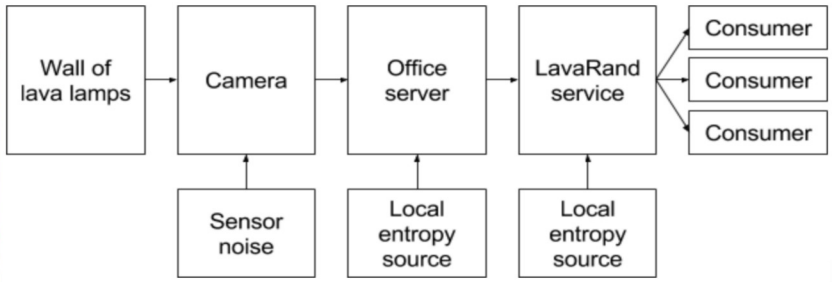
- Hardware RNG lavet af Silicon Graphics
- Udløbet U.S. patent 5.732.138 (1996)
- Et kamera tog et billede af lava lamper som et 140-byte seed til en PRNG
- 1000 byte i sekundet

Hardware

Cloudfare LavaRand

- Cloudfare står for 10% af alt Internet traffik.
- Har mange servere og brug for en masse tilfældige tal.
- Tager et billede af lavalamper 100x100x3 (RGB) bit
- Tilføjer to andre kilder





Hardware

RdRand (tidligere Bull Mountain) fra Intel

- Maskinekode instruktion i nyere Intel og AMD cpu'er
- Tager 2x256 bit entropy samples.
- Køres igennem AES kryptering for at generere 1x256 bit output
- RdRand blev brugt (+ andre kilder) blev brugt på Linux+FreeBSD
- Efter Edward Snowdens afsløringer stoppede Linux+FreeBSD med at bruge RdRand (september 2013)

Hardware

OneRNG

- USB/Serial forbindelse
- Avalanche diode circuit med optionel RF circuit
- Open-source Hardware (OSHW) certificeret
- GPL software
- 10 MB i timen
- 40 USD

Hardware

Infinite Noise TRNG

- USB/Serial forbindelse
- Thermal noise (Johnson–Nyquist støj)
- GPL software
- 10MB i timen
- 35 USD