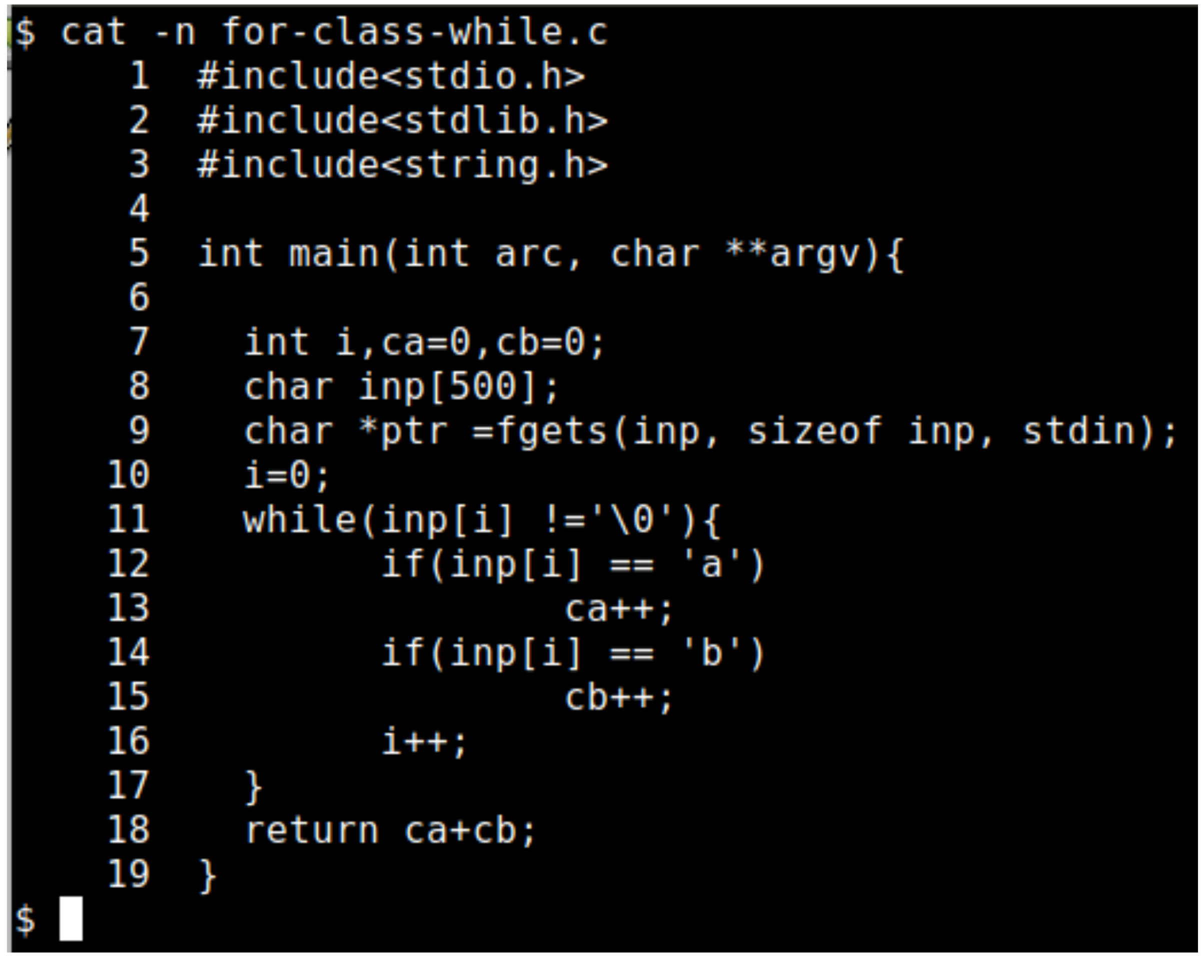
American Fuzzy Lop (AFL)

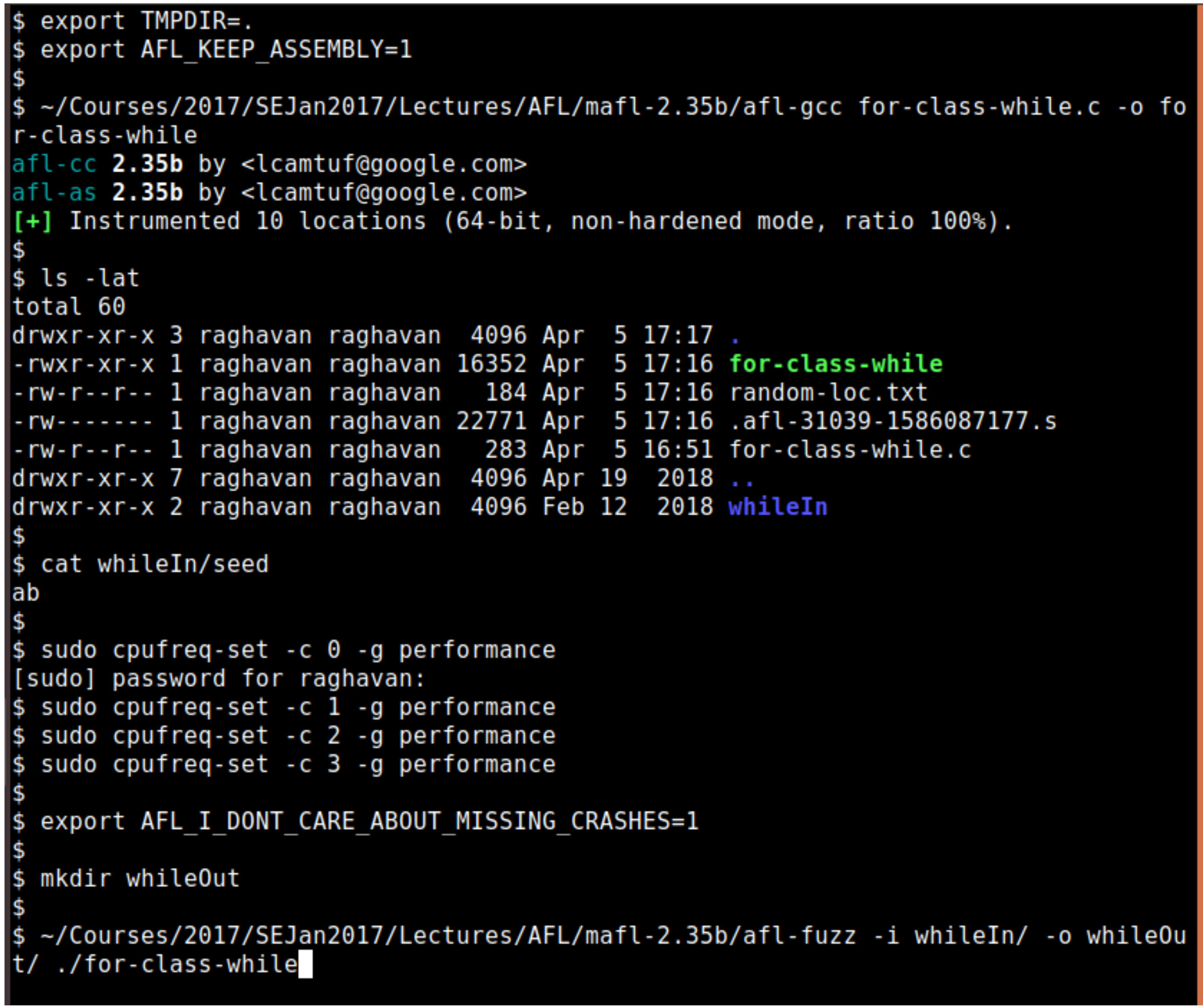
A tool for lightweight automated testing
of C programs

Raghavan Komondoor
Indian Institute of Science, Bangalore

Compiling a program and running AFL on it

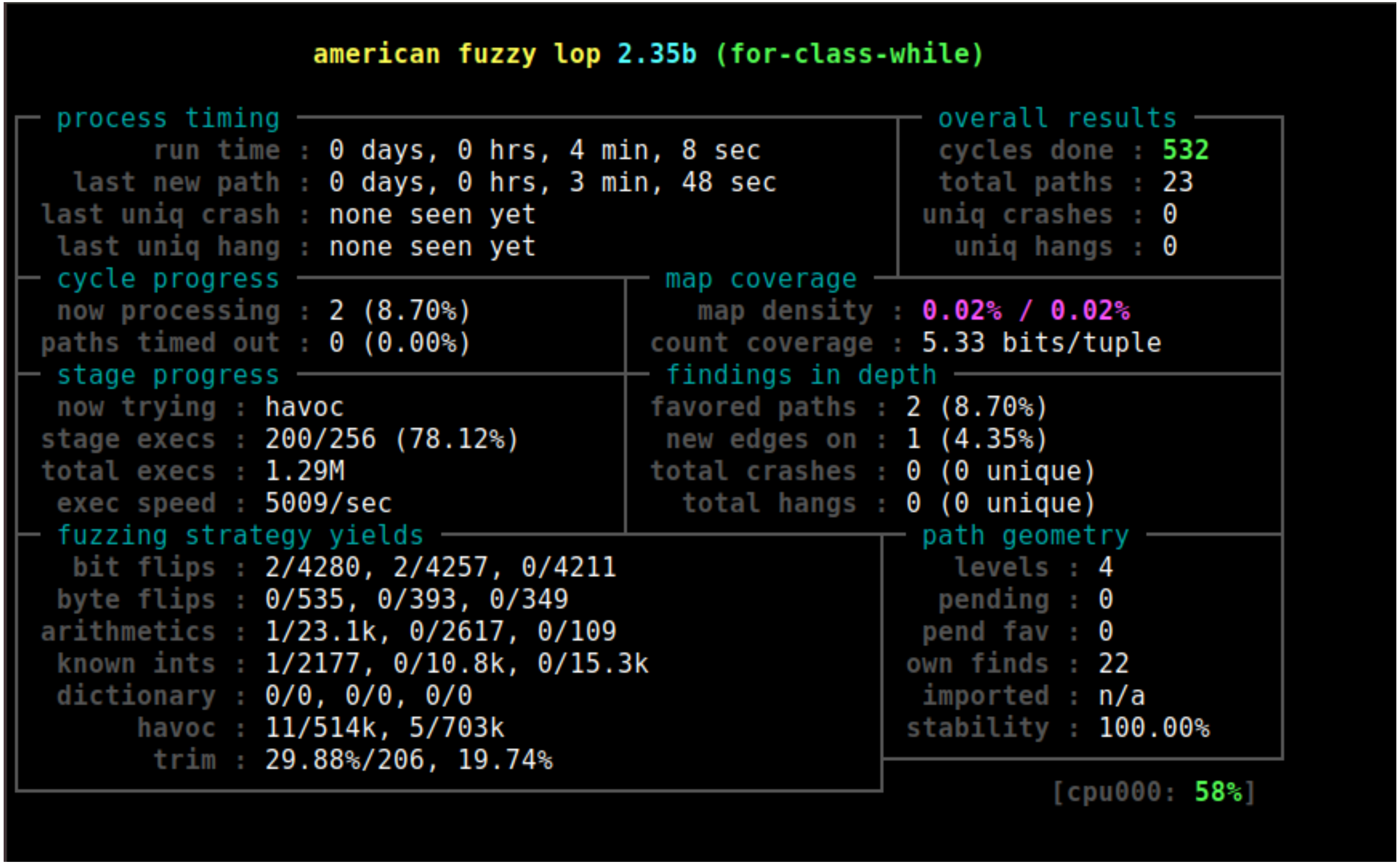


```
$ cat -n for-class-while.c
     1  #include<stdio.h>
     2  #include<stdlib.h>
     3  #include<string.h>
     4
     5  int main(int arc, char **argv){
     6
     7    int i,ca=0,cb=0;
     8    char inp[500];
     9    char *ptr =fgets(inp, sizeof inp, stdin);
    10    i=0;
    11    while(inp[i] !='\0'){
    12          if(inp[i] == 'a')
    13                  ca++;
    14          if(inp[i] == 'b')
    15                  cb++;
    16          i++;
    17    }
    18    return ca+cb;
    19  }
$ 
```



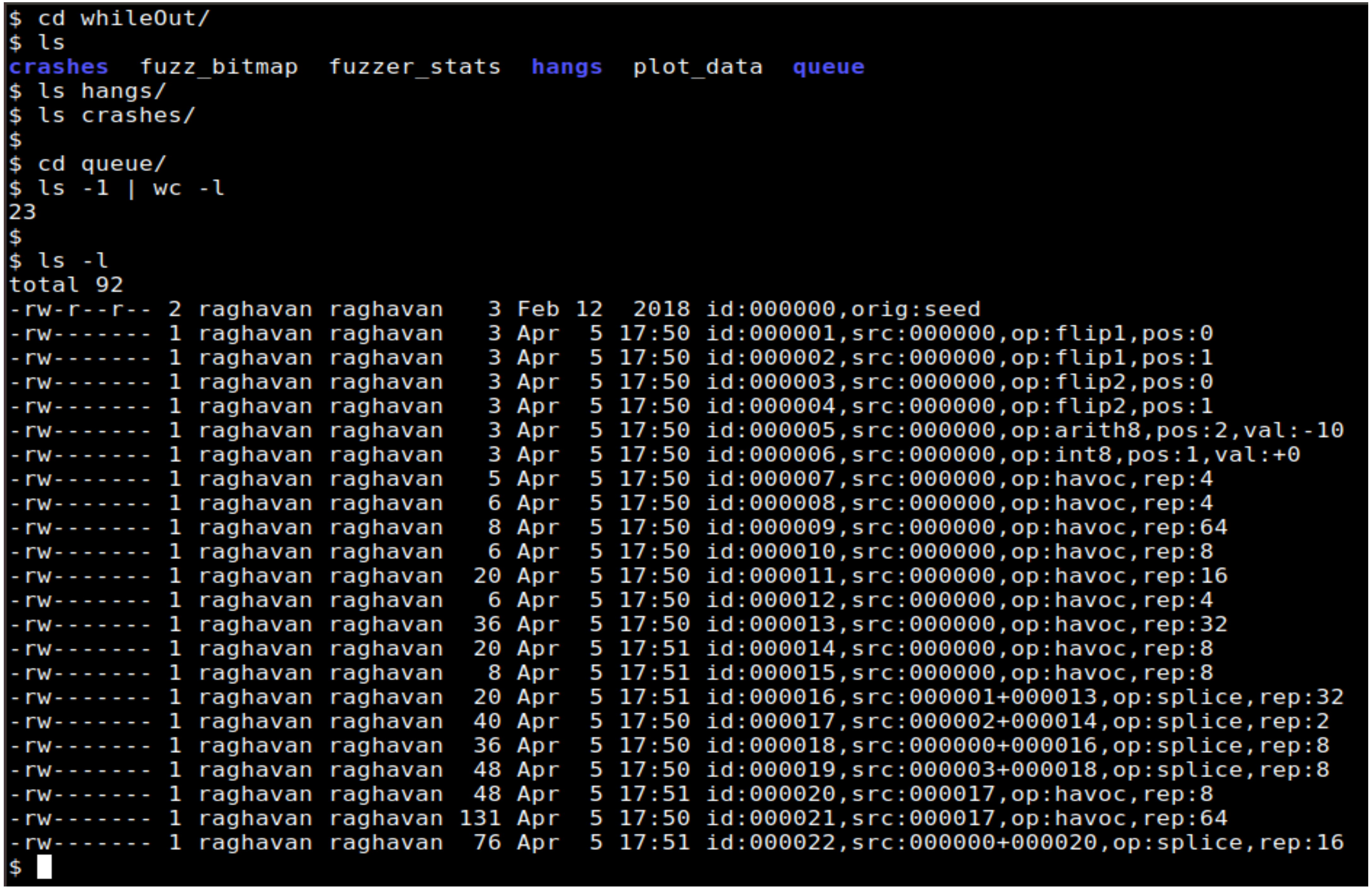
```
$ export TMPDIR=.
$ export AFL_KEEP_ASSEMBLY=1
$
$ ~/Courses/2017/SEJan2017/Lectures/AFL/mafl-2.35b/afl-gcc for-class-while.c -o fo
r-class-while
afl-cc 2.35b by <lcamtuf@google.com>
afl-as 2.35b by <lcamtuf@google.com>
[+] Instrumented 10 locations (64-bit, non-hardened mode, ratio 100%).
$
$ ls -lat
total 60
drwxr-xr-x 3 raghavan raghavan  4096 Apr  5 17:17 .
-rwxr-xr-x 1 raghavan raghavan 16352 Apr  5 17:16 for-class-while
-rw-r--r-- 1 raghavan raghavan   184 Apr  5 17:16 random-loc.txt
-rw------- 1 raghavan raghavan 22771 Apr  5 17:16 .afl-31039-1586087177.s
-rw-r--r-- 1 raghavan raghavan   283 Apr  5 16:51 for-class-while.c
drwxr-xr-x 7 raghavan raghavan  4096 Apr 19  2018 ..
drwxr-xr-x 2 raghavan raghavan  4096 Feb 12  2018 whileIn
$
$ cat whileIn/seed
ab
$
$ sudo cpufreq-set -c 0 -g performance
[sudo] password for raghavan:
$ sudo cpufreq-set -c 1 -g performance
$ sudo cpufreq-set -c 2 -g performance
$ sudo cpufreq-set -c 3 -g performance
$
$ export AFL_I_DONT_CARE_ABOUT_MISSING_CRASHES=1
$
$ mkdir whileOut
$
$ ~/Courses/2017/SEJan2017/Lectures/AFL/mafl-2.35b/afl-fuzz -i whileIn/ -o whileOu
t/ ./for-class-while
```

AFL Run



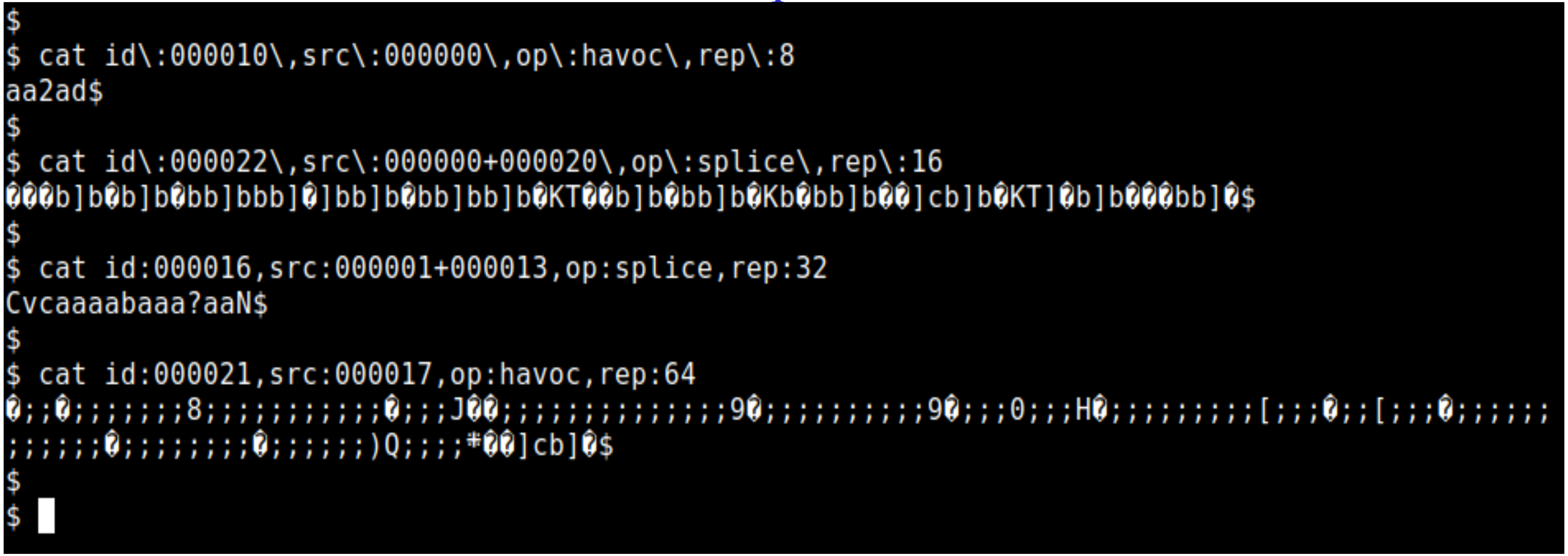
```
                 american fuzzy lop 2.35b (for-class-while)

┌─ process timing ─────────────────────────────────────┬─ overall results ─────┐
│        run time : 0 days, 0 hrs, 4 min, 8 sec         │  cycles done : 532    │
│   last new path : 0 days, 0 hrs, 3 min, 48 sec        │  total paths : 23     │
│ last uniq crash : none seen yet                       │ uniq crashes : 0      │
│  last uniq hang : none seen yet                       │   uniq hangs : 0      │
├─ cycle progress ────────────────┬─ map coverage ─┴───────────────────────┤
│  now processing : 2 (8.70%)      │    map density : 0.02% / 0.02%         │
│ paths timed out : 0 (0.00%)      │ count coverage : 5.33 bits/tuple       │
├─ stage progress ────────────────┼─ findings in depth ────────────────────┤
│  now trying : havoc              │ favored paths : 2 (8.70%)              │
│ stage execs : 200/256 (78.12%)   │  new edges on : 1 (4.35%)              │
│ total execs : 1.29M              │ total crashes : 0 (0 unique)           │
│  exec speed : 5009/sec           │   total hangs : 0 (0 unique)           │
├─ fuzzing strategy yields ───────┴───────────────┬─ path geometry ────────┤
│   bit flips : 2/4280, 2/4257, 0/4211             │    levels : 4          │
│  byte flips : 0/535, 0/393, 0/349                │   pending : 0          │
│ arithmetics : 1/23.1k, 0/2617, 0/109             │  pend fav : 0          │
│  known ints : 1/2177, 0/10.8k, 0/15.3k           │ own finds : 22         │
│  dictionary : 0/0, 0/0, 0/0                      │  imported : n/a        │
│       havoc : 11/514k, 5/703k                    │ stability : 100.00%    │
│        trim : 29.88%/206, 19.74%                 ├────────────────────────┘
└──────────────────────────────────────────────────┘          [cpu000: 58%]
```

AFL Output



```
$ cd whileOut/
$ ls
crashes   fuzz_bitmap   fuzzer_stats   hangs   plot_data   queue
$ ls hangs/
$ ls crashes/
$
$ cd queue/
$ ls -1 | wc -l
23
$
$ ls -l
total 92
-rw-r--r-- 2 raghavan raghavan   3 Feb 12  2018 id:000000,orig:seed
-rw------- 1 raghavan raghavan   3 Apr  5 17:50 id:000001,src:000000,op:flip1,pos:0
-rw------- 1 raghavan raghavan   3 Apr  5 17:50 id:000002,src:000000,op:flip1,pos:1
-rw------- 1 raghavan raghavan   3 Apr  5 17:50 id:000003,src:000000,op:flip2,pos:0
-rw------- 1 raghavan raghavan   3 Apr  5 17:50 id:000004,src:000000,op:flip2,pos:1
-rw------- 1 raghavan raghavan   3 Apr  5 17:50 id:000005,src:000000,op:arith8,pos:2,val:-10
-rw------- 1 raghavan raghavan   3 Apr  5 17:50 id:000006,src:000000,op:int8,pos:1,val:+0
-rw------- 1 raghavan raghavan   5 Apr  5 17:50 id:000007,src:000000,op:havoc,rep:4
-rw------- 1 raghavan raghavan   6 Apr  5 17:50 id:000008,src:000000,op:havoc,rep:4
-rw------- 1 raghavan raghavan   8 Apr  5 17:50 id:000009,src:000000,op:havoc,rep:64
-rw------- 1 raghavan raghavan   6 Apr  5 17:50 id:000010,src:000000,op:havoc,rep:8
-rw------- 1 raghavan raghavan  20 Apr  5 17:50 id:000011,src:000000,op:havoc,rep:16
-rw------- 1 raghavan raghavan   6 Apr  5 17:50 id:000012,src:000000,op:havoc,rep:4
-rw------- 1 raghavan raghavan  36 Apr  5 17:50 id:000013,src:000000,op:havoc,rep:32
-rw------- 1 raghavan raghavan  20 Apr  5 17:51 id:000014,src:000000,op:havoc,rep:8
-rw------- 1 raghavan raghavan   8 Apr  5 17:51 id:000015,src:000000,op:havoc,rep:8
-rw------- 1 raghavan raghavan  20 Apr  5 17:51 id:000016,src:000001+000013,op:splice,rep:32
-rw------- 1 raghavan raghavan  40 Apr  5 17:50 id:000017,src:000002+000014,op:splice,rep:2
-rw------- 1 raghavan raghavan  36 Apr  5 17:50 id:000018,src:000000+000016,op:splice,rep:8
-rw------- 1 raghavan raghavan  48 Apr  5 17:50 id:000019,src:000003+000018,op:splice,rep:8
-rw------- 1 raghavan raghavan  48 Apr  5 17:51 id:000020,src:000017,op:havoc,rep:8
-rw------- 1 raghavan raghavan 131 Apr  5 17:50 id:000021,src:000017,op:havoc,rep:64
-rw------- 1 raghavan raghavan  76 Apr  5 17:51 id:000022,src:000000+000020,op:splice,rep:16
$ 
```

Files in queue folder



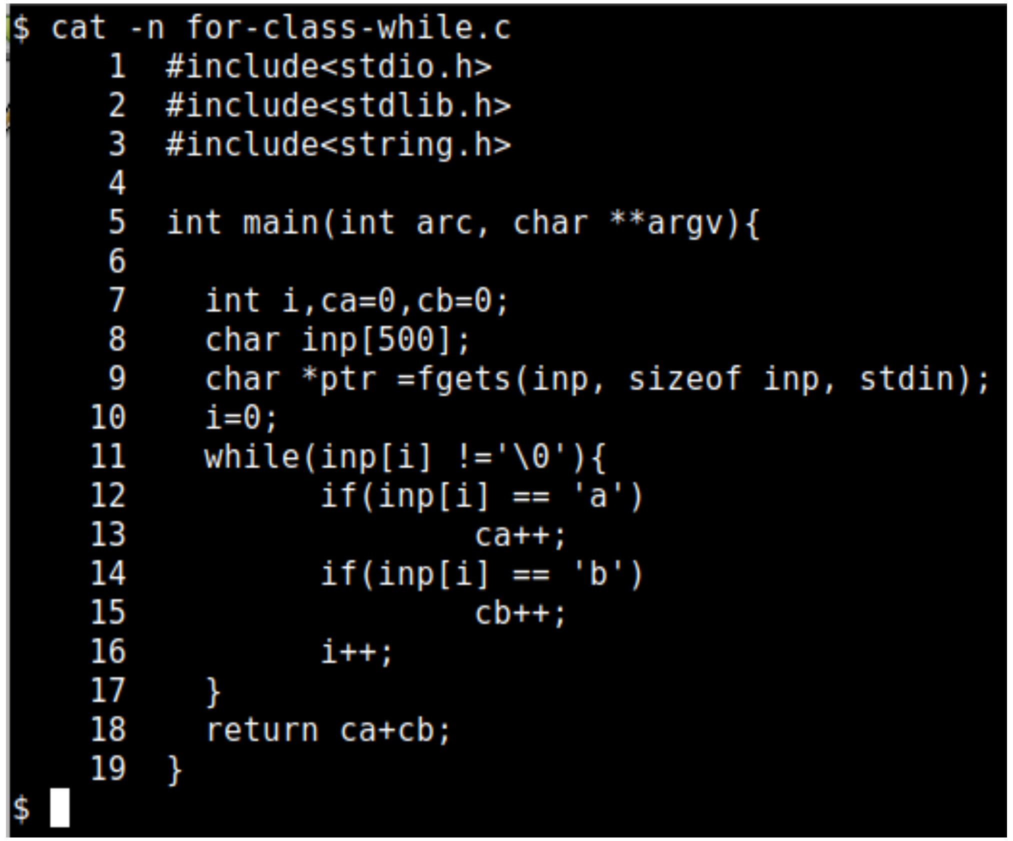
```
$
$ cat id\:000010\,src\:000000\,op\:havoc\,rep\:8
aa2ad$
$
$ cat id\:000022\,src\:000000+000020\,op\:splice\,rep\:16
���b]b�b]b�bb]bbb]�]bb]b�bb]bb]b�KT��b]b�bb]b�Kb�bb]b��]cb]b�KT]�b]b���bb]�$
$
$ cat id:000016,src:000001+000013,op:splice,rep:32
Cvcaaaabaaa?aaN$
$
$ cat id:000021,src:000017,op:havoc,rep:64
�;;�;;;;;;;8;;;;;;;;;;;�;;;J��;;;;;;;;;;;;;;9�;;;;;;;;;;9�;;;0;;;H�;;;;;;;;;[;;;�;;[;;;�;;;;;;
;;;;;;�;;;;;;;;;�;;;;;;)Q;;;;*��]cb]�$
$
$ 
```

AFL aims for branch-pair coverage.

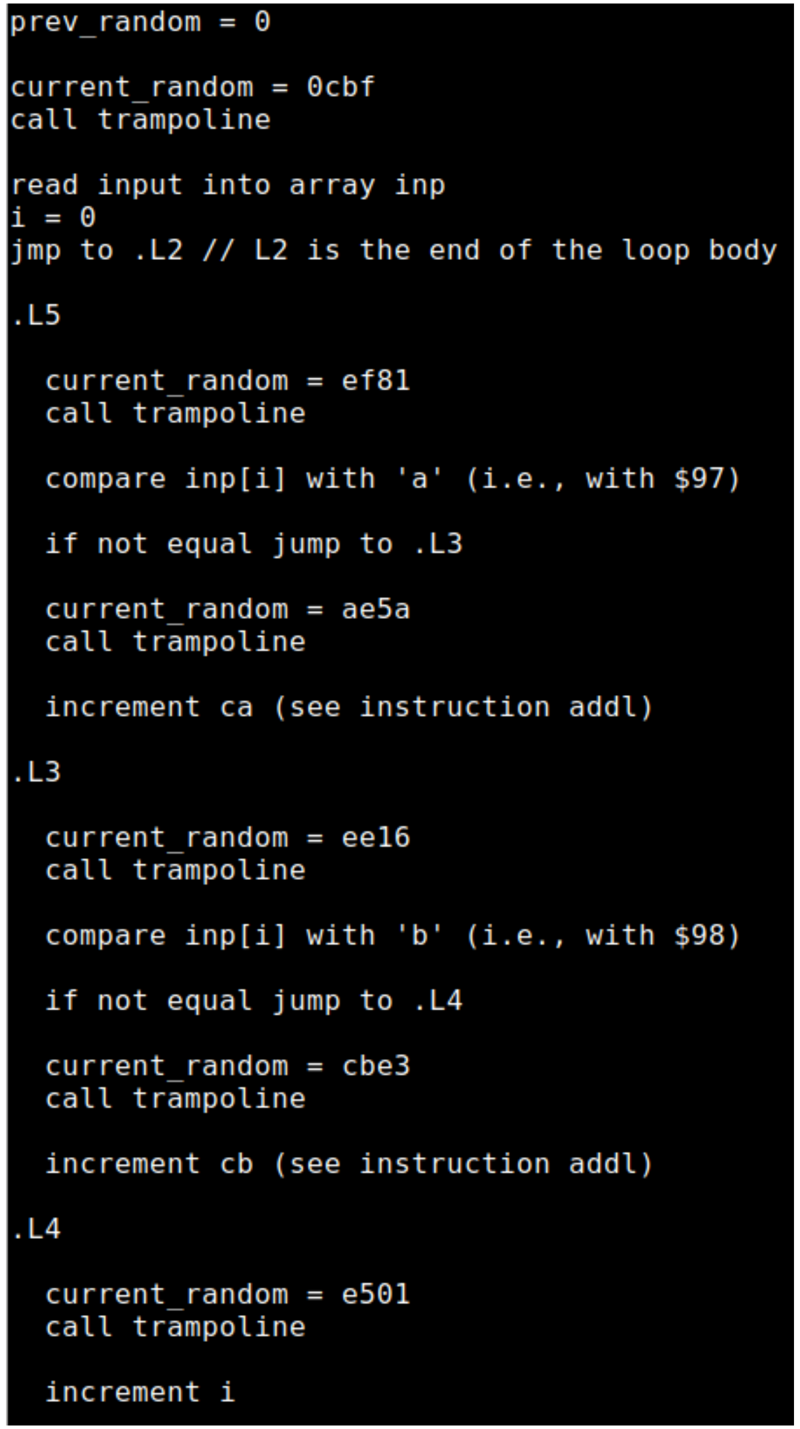
A branch-pair is a sequence of two consecutive branches in the program.

Each input file f that is retained in the queue covers some branch-pair significantly different number of times than all input files that were generated and retained before f.
(Any generated file that does not meet this requirement is thrown away.)

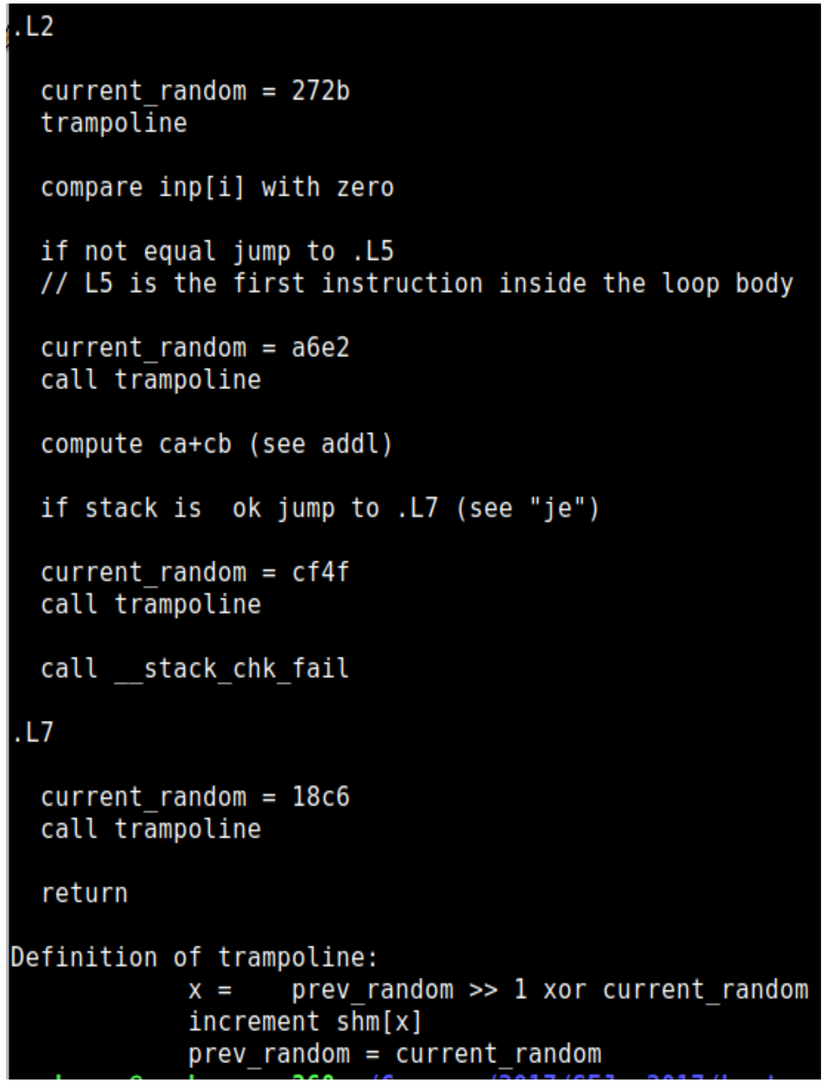
Sample program and outline of its assembly



```
$ cat -n for-class-while.c
     1  #include<stdio.h>
     2  #include<stdlib.h>
     3  #include<string.h>
     4
     5  int main(int arc, char **argv){
     6
     7    int i,ca=0,cb=0;
     8    char inp[500];
     9    char *ptr =fgets(inp, sizeof inp, stdin);
    10    i=0;
    11    while(inp[i] !='\0'){
    12          if(inp[i] == 'a')
    13                  ca++;
    14          if(inp[i] == 'b')
    15                  cb++;
    16          i++;
    17    }
    18    return ca+cb;
    19  }
$ 
```

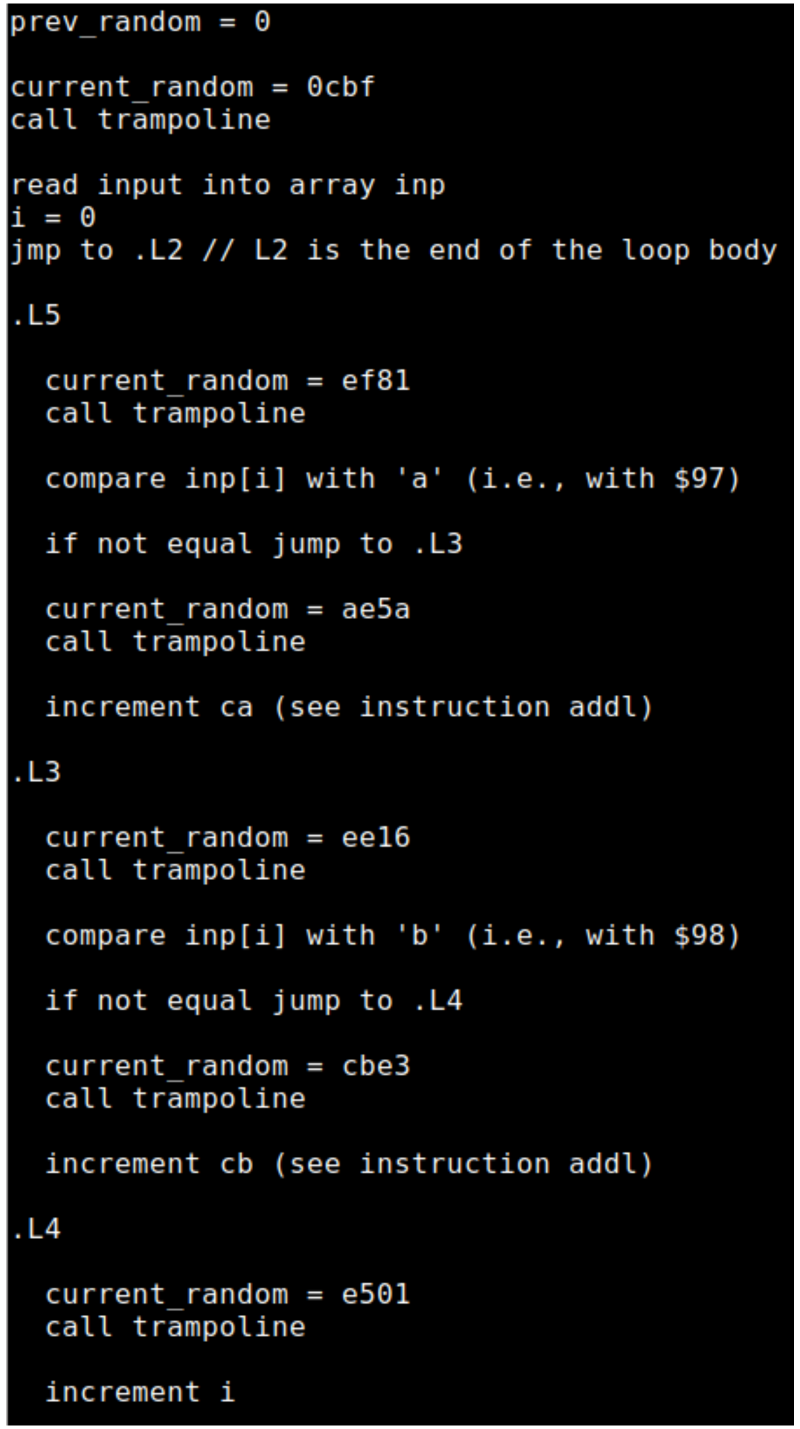


```
prev_random = 0

current_random = 0cbf
call trampoline

read input into array inp
i = 0
jmp to .L2 // L2 is the end of the loop body

.L5

  current_random = ef81
  call trampoline

  compare inp[i] with 'a' (i.e., with $97)

  if not equal jump to .L3

  current_random = ae5a
  call trampoline

  increment ca (see instruction addl)

.L3

  current_random = ee16
  call trampoline

  compare inp[i] with 'b' (i.e., with $98)

  if not equal jump to .L4

  current_random = cbe3
  call trampoline

  increment cb (see instruction addl)

.L4

  current_random = e501
  call trampoline

  increment i
```

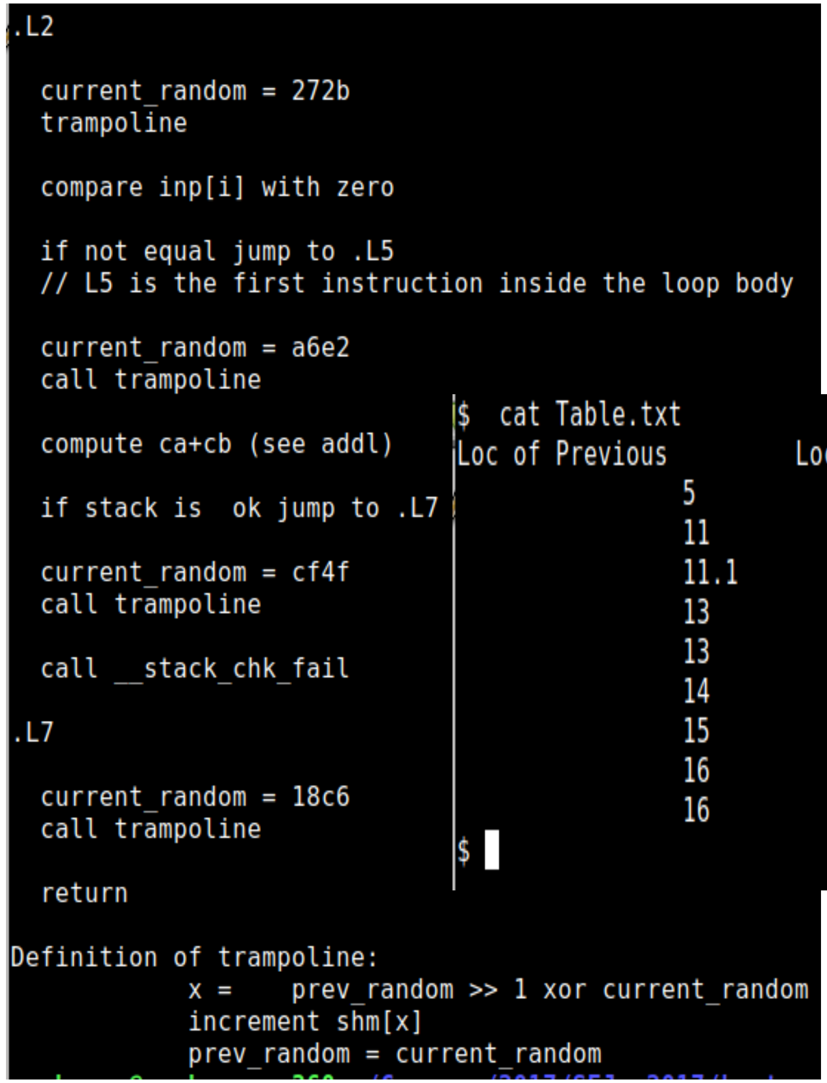


```
.L2

  current_random = 272b
  trampoline

  compare inp[i] with zero

  if not equal jump to .L5
  // L5 is the first instruction inside the loop body

  current_random = a6e2
  call trampoline

  compute ca+cb (see addl)

  if stack is  ok jump to .L7 (see "je")

  current_random = cf4f
  call trampoline

  call __stack_chk_fail

.L7

  current_random = 18c6
  call trampoline

  return

Definition of trampoline:
           x =    prev_random >> 1 xor current_random
           increment shm[x]
           prev_random = current_random
```

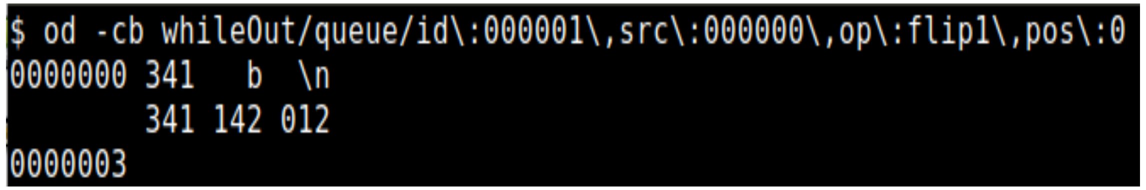
Branch-pair profile of a run



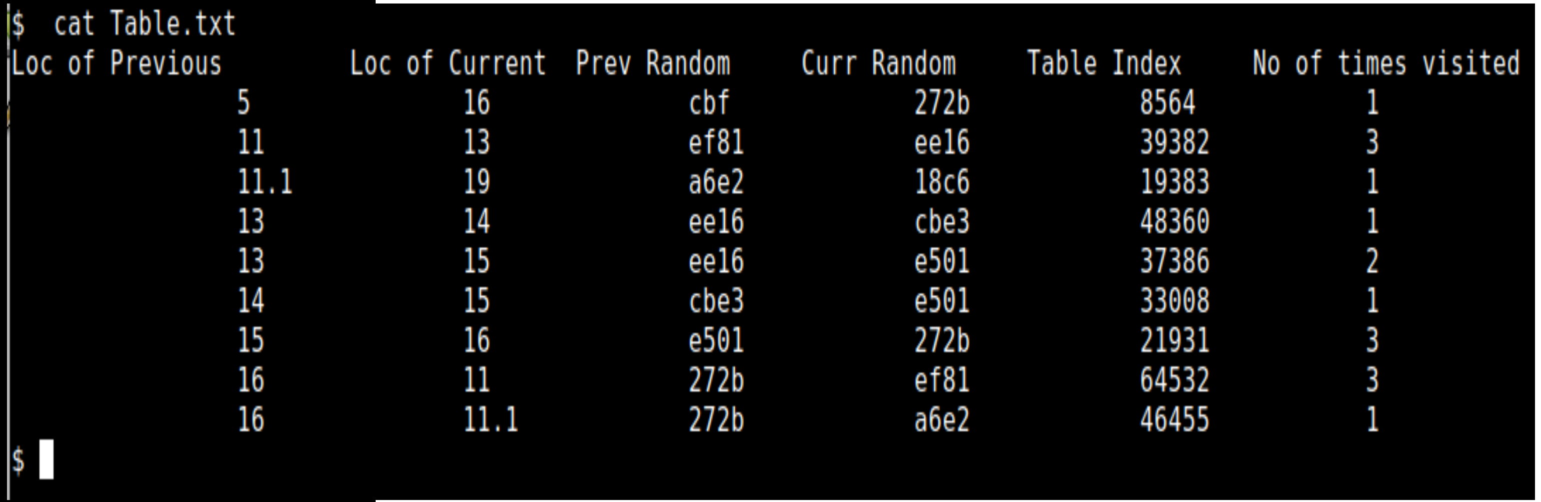
```
prev_random = 0

current_random = 0cbf
call trampoline

read input into array inp
i = 0
jmp to .L2 // L2 is the end of the loop body

.L5

  current_random = ef81
  call trampoline

  compare inp[i] with 'a' (i.e., with $97)

  if not equal jump to .L3

  current_random = ae5a
  call trampoline

  increment ca (see instruction addl)

.L3

  current_random = ee16
  call trampoline

  compare inp[i] with 'b' (i.e., with $98)

  if not equal jump to .L4

  current_random = cbe3
  call trampoline

  increment cb (see instruction addl)

.L4

  current_random = e501
  call trampoline

  increment i
```



```
.L2

  current_random = 272b
  trampoline

  compare inp[i] with zero

  if not equal jump to .L5
  // L5 is the first instruction inside the loop body

  current_random = a6e2
  call trampoline

  compute ca+cb (see addl)

  if stack is  ok jump to .L7

  current_random = cf4f
  call trampoline

  call __stack_chk_fail

.L7

  current_random = 18c6
  call trampoline

  return

Definition of trampoline:
            x =    prev_random >> 1 xor current_random
            increment shm[x]
            prev_random = current_random
```

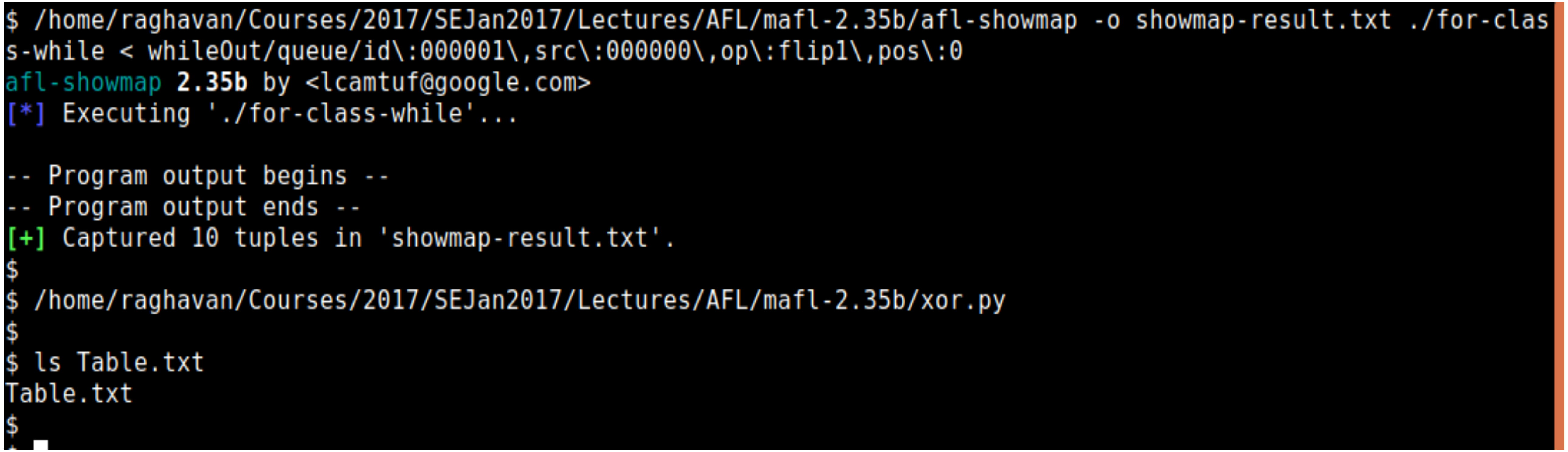


```
$ od -cb whileOut/queue/id\:000001\,src\:000000\,op\:flip1\,pos\:0
0000000 341   b  \n
        341 142 012
0000003
```



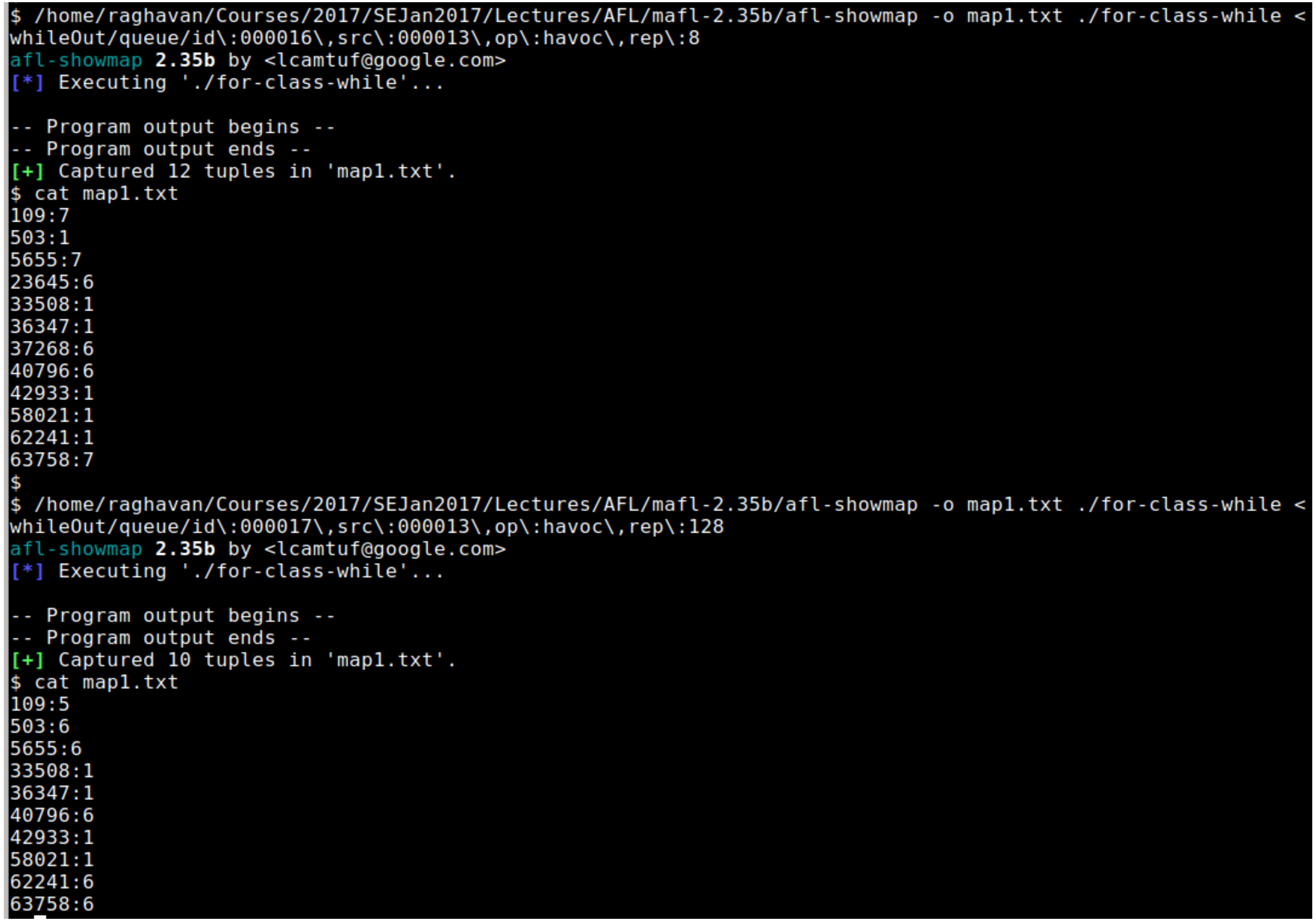
```
$  cat Table.txt
Loc of Previous         Loc of Current  Prev Random     Curr Random     Table Index     No of times visited
                5               16              cbf             272b            8564            1
                11              13              ef81            ee16            39382           3
                11.1            19              a6e2            18c6            19383           1
                13              14              ee16            cbe3            48360           1
                13              15              ee16            e501            37386           2
                14              15              cbe3            e501            33008           1
                15              16              e501            272b            21931           3
                16              11              272b            ef81            64532           3
                16              11.1            272b            a6e2            46455           1
$ 
```

Steps to extract branch-pair profile of a run



```
$ /home/raghavan/Courses/2017/SEJan2017/Lectures/AFL/mafl-2.35b/afl-showmap -o showmap-result.txt ./for-clas
s-while < whileOut/queue/id\:000001\,src\:000000\,op\:flip1\,pos\:0
afl-showmap 2.35b by <lcamtuf@google.com>
[*] Executing './for-class-while'...

-- Program output begins --
-- Program output ends --
[+] Captured 10 tuples in 'showmap-result.txt'.
$
$ /home/raghavan/Courses/2017/SEJan2017/Lectures/AFL/mafl-2.35b/xor.py
$
$ ls Table.txt
Table.txt
$
```

Branch-pair profiles of different input files



```
$ /home/raghavan/Courses/2017/SEJan2017/Lectures/AFL/mafl-2.35b/afl-showmap -o map1.txt ./for-class-while <
whileOut/queue/id\:000016\,src\:000013\,op\:havoc\,rep\:8
afl-showmap 2.35b by <lcamtuf@google.com>
[*] Executing './for-class-while'...

-- Program output begins --
-- Program output ends --
[+] Captured 12 tuples in 'map1.txt'.
$ cat map1.txt
109:7
503:1
5655:7
23645:6
33508:1
36347:1
37268:6
40796:6
42933:1
58021:1
62241:1
63758:7
$
$ /home/raghavan/Courses/2017/SEJan2017/Lectures/AFL/mafl-2.35b/afl-showmap -o map1.txt ./for-class-while <
whileOut/queue/id\:000017\,src\:000013\,op\:havoc\,rep\:128
afl-showmap 2.35b by <lcamtuf@google.com>
[*] Executing './for-class-while'...

-- Program output begins --
-- Program output ends --
[+] Captured 10 tuples in 'map1.txt'.
$ cat map1.txt
109:5
503:6
5655:6
33508:1
36347:1
40796:6
42933:1
58021:1
62241:6
63758:6
```