

## Introduction to Operating Systems

### Project 1B: Perf tool patch

This document shows how to patch the **perf** tool source code to successfully build it.

#### Affected kernels:

- 4.19.149
- 4.19.150
- 4.19.151
- 4.19.152

#### Problem:

A problem in the source code prevents to successfully build the tool.

```
user@ubuntu:~/usr/src/linux-4.19.149/tools/perf$ make
LINK      plugin_cscli.so
LINK      plugin_cfg80211.so
CC        builtin-bench.o
CC        builtin-annotate.o
CC        builtin-config.o
CC        builtin-dtff.o
CC        builtin-evlist.o
CC        util/block-range.o
CC        builtin-fttrace.o
CC        util/build-id.o
CC        builtin-help.o
CC        builtin-sched.o
CC        util/config.o
CC        util/ctype.o
CC        util/db-export.o
CC        util/env.o
CC        util/event.o
CC        builtin-buildid-list.o
CC        builtin-buildid-cache.o
CC        util/evlist.o
CC        builtin-kallsyms.o
CC        builtin-list.o
CC        builtin-record.o
CC        util/evsel.o
In file included from util/evlist.h:15:0,
                 from util/evsel.c:30:
util/evsel.c: In function 'perf_evsel_exit':
util/evsel.c:25:28: error: passing argument 1 of 'free' discards 'const' qualifier
r from pointer target type [-Werror=discarded-qualifiers]
#define zfree(ptr) ((free)(ptr); *ptr = NULL; })
                        ^
util/evsel.c:1293:2: note: in expansion of macro 'zfree'
  zfree(&evsel->pmu_name);
  ^
In file included from /usr/src/linux-4.19.149/tools/perf/arch/x86/include/perf_
egs.h:5:0,
                 from util/perf_regs.h:27,
                 from util/event.h:11,
                 from util/callchain.h:8,
                 from util/evsel.c:26:
/usr/include/stdint.h:483:13: note: expected 'void *' but argument is of type 'c
onst char *'
extern void free (void * __ptr) __THROW;
                  ^
CC        builtin-report.o
CC        builtin-stat.o
cc1: all warnings being treated as errors
/usr/src/linux-4.19.149/tools/build/Makefile.build:96: recipe for target 'util/e
vsel.o' failed
make[4]: *** [util/evsel.o] Error 1
/usr/src/linux-4.19.149/tools/build/Makefile.build:139: recipe for target 'util'
failed
make[3]: *** [util] Error 2
Makefile.perf:633: recipe for target 'libperf-in.o' failed
```

#### Git comments:

This problem was discovered by the official developers on Sept. 29-30/2020:

- <https://lkml.org/lkml/2020/9/29/2330>
- <https://lkml.org/lkml/2020/9/30/1024>

## Solutions:

There are two official solutions for this problem:

1. Use kernel 4.19.148:

<https://cdn.kernel.org/pub/linux/kernel/v4.x/linux-4.19.148.tar.xz>

This solution requires Section 2 and Section 3 of project 1A.

2. In the affected kernels, go to **linux-4.19.149/tools/perf/util/evsel.c**.

Locate the following function:

**void perf\_evsel\_\_exit(struct perf\_evsel \*evsel)**

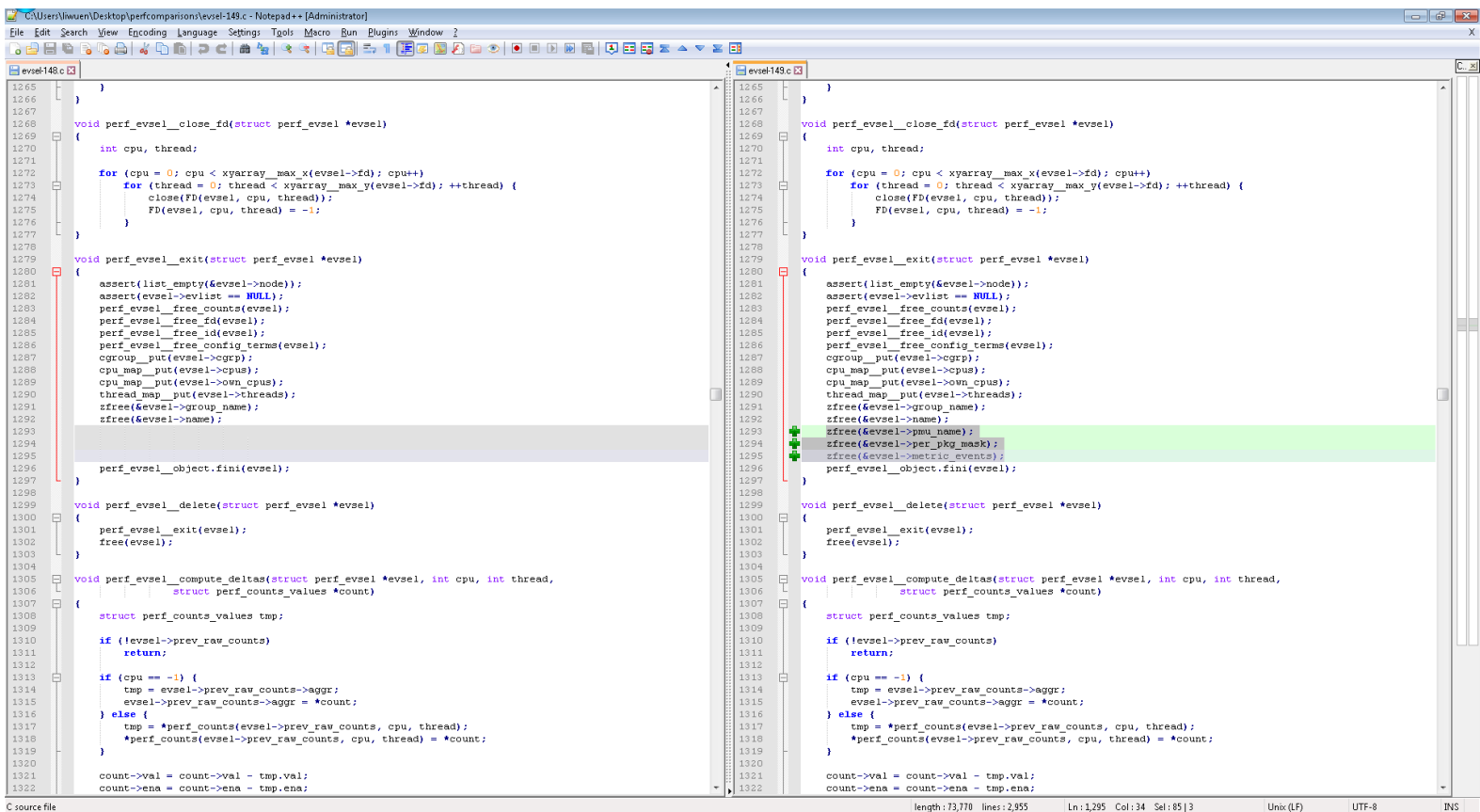
and comment the lines:

**zfree(&evsel->pmu\_name);**

**zfree(&evsel->per\_pkg\_mask);**

**zfree(&evsel->metric\_events);**

(Around lines 1293, 1294 and 1295).



```
1265 }
1266 }
1267
1268 void perf_evsel_close_fd(struct perf_evsel *evsel)
1269 {
1270     int cpu, thread;
1271
1272     for (cpu = 0; cpu < xyarray_max_x(evsel->fd); cpu++)
1273         for (thread = 0; thread < xyarray_max_y(evsel->fd); ++thread) {
1274             close(fd(evsel, cpu, thread));
1275             FD(evsel, cpu, thread) = -1;
1276         }
1277 }
1278
1279 void perf_evsel__exit(struct perf_evsel *evsel)
1280 {
1281     assert(list_empty(&evsel->nnode));
1282     assert(evsel->evlist == NULL);
1283     perf_evsel_free_counts(evsel);
1284     perf_evsel_free_fd(evsel);
1285     perf_evsel_free_id(evsel);
1286     perf_evsel_free_config_terms(evsel);
1287     cgroup_put(evsel->cgrp);
1288     cpu_map_put(evsel->cpus);
1289     cpu_map_put(evsel->own_cpus);
1290     thread_map_put(evsel->threads);
1291     zfree(&evsel->group_name);
1292     zfree(&evsel->name);
1293     // zfree(&evsel->pmu_name);
1294     // zfree(&evsel->per_pkg_mask);
1295     // zfree(&evsel->metric_events);
1296     perf_evsel_object.fini(evsel);
1297 }
1298
1299 void perf_evsel_delete(struct perf_evsel *evsel)
1300 {
1301     perf_evsel__exit(evsel);
1302     free(evsel);
1303 }
1304
1305 void perf_evsel_compute_deltas(struct perf_evsel *evsel, int cpu, int thread,
1306                               struct perf_counts_values *count)
1307 {
1308     struct perf_counts_values tmp;
1309
1310     if (!evsel->prev_raw_counts)
1311         return;
1312
1313     if (cpu == -1) {
1314         tmp = evsel->prev_raw_counts->aggr;
1315         evsel->prev_raw_counts->aggr = *count;
1316     } else {
1317         tmp = *perf_counts(evsel->prev_raw_counts, cpu, thread);
1318         *perf_counts(evsel->prev_raw_counts, cpu, thread) = *count;
1319     }
1320
1321     count->val = count->val - tmp.val;
1322     count->ena = count->ena - tmp.ena;
```

After commenting these lines, save the evsel.c file, and run the commands

**\$ sudo make**

**\$ sudo make install**

(Please refer to Section 2.1 Project 1B)

After patching this file, the build process should be successful.

```
LD      util/libperf-in.o
LD      libperf-in.o
AR      libperf.a
LINK    perf
INSTALL tests
INSTALL binaries
INSTALL libexec
INSTALL bpf-headers
INSTALL bpf-examples
INSTALL perf-archive
INSTALL perf-with-kcore
INSTALL strace/groups
INSTALL perf_completion-script
INSTALL perf-tip
usertest@ubuntu:/usr/src/linux-4.19.149/tools/perf$ perf --version
perf version 4.19.149
usertest@ubuntu:/usr/src/linux-4.19.149/tools/perf$ perf
usage: perf [--version] [--help] [OPTIONS] COMMAND [ARGS]

The most commonly used perf commands are:
  annotate      Read perf.data (created by perf record) and display annotated code
  archive      Create archive with object files with build-ids found in perf.data file
  bench        General framework for benchmark suites
  buildid-cache Manage build-id cache.
  buildid-list List the buildids in a perf.data file
  c2c          Shared Data C2C/HITM Analyzer.
  config       Get and set variables in a configuration file.
  data        Data file related processing
  diff        Read perf.data files and display the differential profile
  evlist      List the event names in a perf.data file
  ftrace      Simple wrapper for kernel's ftrace functionality
  inject      Filter to augment the events stream with additional information
  kallsyms    Searches running kernel for symbols
  knmem       Tool to trace/measure kernel memory properties
  kvm         Tool to trace/measure kvm guest os
  list        List all symbolic event types
  lock        Analyze lock events
  mem         Profile memory accesses
  record      Run a command and record its profile into perf.data
  report      Read perf.data (created by perf record) and display the profile
  sched       Tool to trace/measure scheduler properties (latencies)
  script      Read perf.data (created by perf record) and display trace output
  stat        Run a command and gather performance counter statistics
  test        Runs sanity tests.
  tui         Tool to visualize total system behavior during a workload
  top         System profiling tool.
  probe       Define new dynamic tracepoints
  trace       strace inspired tool

See 'perf help COMMAND' for more information on a specific command.
usertest@ubuntu:/usr/src/linux-4.19.149/tools/perf$ cd ~/Desktop/
usertest@ubuntu:~/Desktop$ ls
Profiling tests
usertest@ubuntu:~/Desktop$ cd Profiling\ tests/
usertest@ubuntu:~/Desktop/Profiling tests$ ls
```

## IMPORTANT NOTES:

1. This patch is a **best-effort** solution (so no guarantees that this will solve all the possible problems).  
If more problems are found later in the affected kernels, please use kernel 4.19.148.
2. In order to avoid this situation in the future, **Project 2 and Project 3 will require to use kernel 4.19.148** (so you will have to download and compile kernel 4.19.148).
3. For more information, please check the related post on E3 forum:

<https://e3new.nctu.edu.tw/mod/forum/discuss.php?d=72257>