軟體分析與最佳化 HW5

612410017 林靖紳

Execution environments

- CPU information

```
ashen@Stephanie-Lin:~$ lscpu
Architecture:            x86_64
  CPU op-mode(s):        32-bit, 64-bit
  Address sizes:         39 bits physical, 48 bits virtual
  Byte Order:            Little Endian
CPU(s):                  12
  On-line CPU(s) list:   0-11
Vendor ID:               GenuineIntel
  Model name:            11th Gen Intel(R) Core(TM) i5-11500 @ 2.70GHz
    CPU family:          6
    Model:               167
    Thread(s) per core:  2
    Core(s) per socket:  6
    Socket(s):           1
    Stepping:            1
    CPU max MHz:         4600.0000
    CPU min MHz:         800.0000
    BogoMIPS:            5424.00
```

- Memory

```
ashen@Stephanie-Lin:~$ free -h
               total        used        free      shared  buff/cache   available
Mem:            31Gi       4.2Gi        12Gi       1.8Gi        14Gi        24Gi
Swap:          2.0Gi          0B       2.0Gi
```

- OS version

```
ashen@Stephanie-Lin:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 22.04.2 LTS
Release:        22.04
Codename:       jammy
```

- GCC version

```
ashen@Stephanie-Lin:~$ gcc --version
gcc (Ubuntu 11.4.0-1ubuntu1~22.04) 11.4.0
Copyright (C) 2021 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

- ICC version

```
ashen@Stephanie-Lin:~/Documents/Software_Analysis-git/HW2$ icc --version
icc: remark #10441: The Intel(R) C++ Compiler Classic (ICC) is deprecated and will be removed from
product release in the second half of 2023. The Intel(R) oneAPI DPC++/C++ Compiler (ICX) is the rec
ommended compiler moving forward. Please transition to use this compiler. Use '-diag-disable=10441'
 to disable this message.
icc (ICC) 2021.10.0 20230609
Copyright (C) 1985-2023 Intel Corporation.  All rights reserved.
```

- gcov version

```
ashen@Stephanie-Lin:~$ gcov --version
gcov (Ubuntu 11.4.0-1ubuntu1~22.04) 11.4.0
```

- lcov version

```
ashen@Stephanie-Lin:~$ lcov --version
lcov: LCOV version 1.14
```

Q1: 根據gcov的分析結果，請問程式的執行覆蓋率是多少?

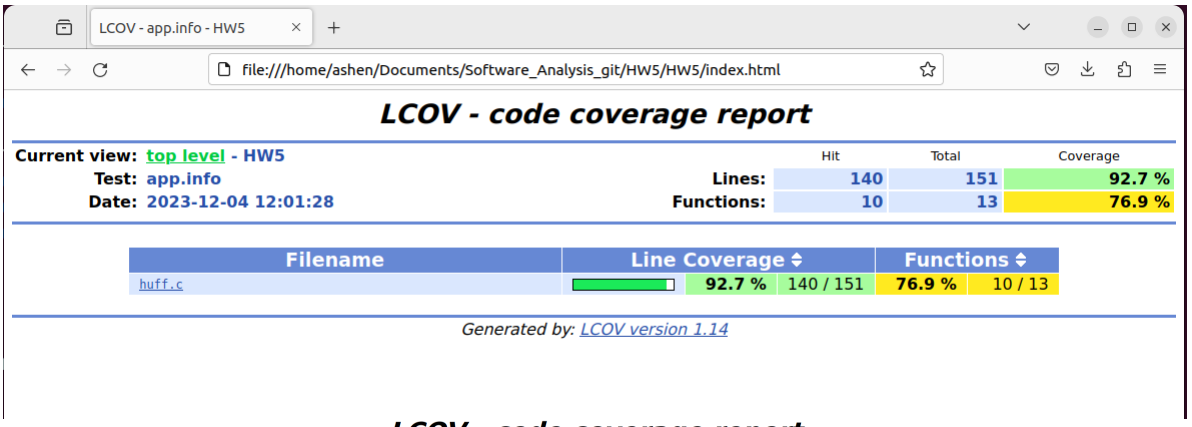
- 92.72 %

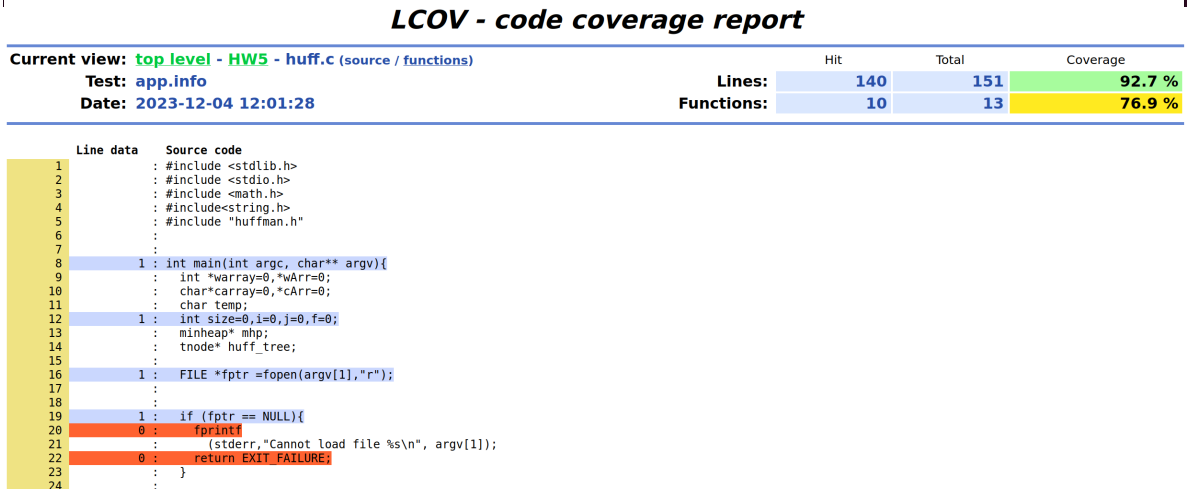
```
ashen@Stephanie-Lin:~/Documents/Software_Analysis_git/HW5$ gcc -fprofile-arcs -ftest-coverage -O2 huff.c -o huff_exe
ashen@Stephanie-Lin:~/Documents/Software_Analysis_git/HW5$ ./huff_exe input.data
ashen@Stephanie-Lin:~/Documents/Software_Analysis_git/HW5$ gcov huff_exe-huff.gcno
File 'huff.c'
Lines executed:92.72% of 151
Creating 'huff.c.gcov'
```

- lcov

```
ashen@Stephanie-Lin:~/Documents/Software_Analysis_git/HW5$ lcov --directory . --capture --output-file app.info
Capturing coverage data from .
Subroutine read_intermediate_text redefined at /usr/bin/geninfo line 2623.
Subroutine read_intermediate_json redefined at /usr/bin/geninfo line 2655.
Subroutine intermediate_text_to_info redefined at /usr/bin/geninfo line 2703.
Subroutine intermediate_json_to_info redefined at /usr/bin/geninfo line 2792.
Subroutine get_output_fd redefined at /usr/bin/geninfo line 2872.
Subroutine print_gcov_warnings redefined at /usr/bin/geninfo line 2900.
Subroutine process_intermediate redefined at /usr/bin/geninfo line 2930.
Found gcov version: 11.4.0
Using intermediate gcov format
Scanning . for .gcda files ...
Found 1 data files in .
Processing huff_exe-huff.gcda
Finished .info-file creation
ashen@Stephanie-Lin:~/Documents/Software_Analysis_git/HW5$ genhtml app.info
Reading data file app.info
Found 2 entries.
Found common filename prefix "/home/ashen/Documents/Software_Analysis_git"
Writing .css and .png files.
Generating output.
Processing file HW5/huff.c
```

- lcov 產生的 html





Q2: huff.c中的134與209行是否有被執行? 分別被執行幾次?

- line 134: 有被執行，執行 740 次

```
133        740 :   if(right<mhp->size && (mhp->array[right]->weight) < (mhp->array[min]->weight)){
134            :     min = right;
135            :   }
```

- line 209：有備執行，執行 2469604 次

```
208    3974483 :           if(sequence[j]==0){
209    2469604 :             write_bits(bit_fld,0,cnt,fptr);
210            :           }
```

Q3: 請問你覺得使用 gcov & lcov，與 intel codecov 的差異與優缺點?

- **差異：**
  - 支援的架構：
    - gcov & lcov： 通常與 GNU 一起使用，主要針對通用的GNU/Linux 環境和其他 Unix-like 系統
    - Intel codecov： 專為 Intel 架構（如x86和x86-64）而設計
  - 實現：
    - gcov & lcov： 使用軟體插桿，在編譯時使用 -fprofile-arcs -ftest-coverage 選項，並通過解析執行時的數據（使用gcov）來獲取覆蓋率等資訊。
    - Intel codecov： 利用硬體特性，例如 Intel Processor Trace (IPT) 和 Pin tool，來獲取指令和程式碼執行的細節。
  - 工具特點：
    - gcov & lcov：
      - 與GNU工具鏈整合良好。
      - 生成HTML報告以直觀展示覆蓋率等資訊。
      - 適用於通用的 GNU/Linux 和 Unix-like 環境。
    - Intel Code Coverage：
      - 專為 Intel 架構優化，利用硬體特性進行覆蓋率分析。
      - 與 Intel 的其他工具集成，如 VTune Profiler。
- **優缺點：**
  - gcov & lcov：
    - 優點：易於使用，廣泛支持，與GNU工具鏈整合，適用於通用環境。
    - 缺點：在某些情況下，可能需要手動配置和調整。
  - Intel Code Coverage：
    - 優點： 利用硬體特性，提供更深入的性能分析，與 Intel 工具集成。
    - 缺點： 對於非 Intel 架構不適用，需要特定硬體支持。