

Open

outp
~/Desktop

Save

1 Flat profile:

2

3 Each sample counts as 0.01 seconds.

4 no time accumulated

5

6 % cumulative self self total

7 time seconds seconds calls Ts/call Ts/call name

8 0.00 0.00 0.00 7 0.00 0.00 swap(int*, int*)

9 0.00 0.00 0.00 4 0.00 0.00 partition(int*, int, int)

10 0.00 0.00 0.00 2 0.00 0.00 printArray(int*, int)

11 0.00 0.00 0.00 1 0.00 0.00 __static_initialization_and_destruction_0(int, int)

12 0.00 0.00 0.00 1 0.00 0.00 quickSort(int*, int, int)

13

14 % the percentage of the total running time of the

15 time program used by this function.

16

17 cumulative a running sum of the number of seconds accounted

18 seconds for by this function and those listed above it.

19

20 self the number of seconds accounted for by this

21 seconds function alone. This is the major sort for this

22 listing.

23

24 calls the number of times this function was invoked, if

25 this function is profiled, else blank.

26

27 self the average number of milliseconds spent in this

28 ms/call function per call, if this function is profiled,

29 else blank.

30

31 total the average number of milliseconds spent in this

32 ms/call function and its descendents per call, if this

33 function is profiled, else blank.

34

35 name the name of the function. This is the minor sort

36 for this listing. The index shows the location of

37 the function in the gprof listing. If the index is

Plain Text

Tab Width: 7

Ln 1, Col 1

INS

Open

outp
~/Desktop

Save

39 the gprof listing if it were to be printed.

40

41 Copyright (C) 2012-2022 Free Software Foundation, Inc.

42

43 Copying and distribution of this file, with or without modification,

44 are permitted in any medium without royalty provided the copyright

45 notice and this notice are preserved.

46

47 Call graph (explanation follows)

48

49

50 granularity: each sample hit covers 4 byte(s) no time propagated

51

52 index % time self children called name

53 0.00 0.00 0.00 7/7 partition(int*, int, int) [9]

54 [8] 0.0 0.00 0.00 7 swap(int*, int*) [8]

55 -----

56 0.00 0.00 0.00 4/4 quickSort(int*, int, int) [12]

57 [9] 0.0 0.00 0.00 4 partition(int*, int, int) [9]

58 0.00 0.00 0.00 7/7 swap(int*, int*) [8]

59 -----

60 0.00 0.00 0.00 2/2 main [6]

61 [10] 0.0 0.00 0.00 2 printArray(int*, int) [10]

62 -----

63 0.00 0.00 0.00 1/1 _GLOBAL_sub_I_Z4swapPiS_ [13]

64 [11] 0.0 0.00 0.00 1 __static_initialization_and_destruction_0(int, int) [11]

65 -----

66 0.00 0.00 0.00 8 quickSort(int*, int, int) [12]

67 0.00 0.00 0.00 1/1 main [6]

68 [12] 0.0 0.00 0.00 1+8 quickSort(int*, int, int) [12]

69 0.00 0.00 0.00 4/4 partition(int*, int, int) [9]

70 8 quickSort(int*, int, int) [12]

71 -----

72

73 This table describes the call tree of the program, and was sorted by

74 the total amount of time spent in each function and its children.

75

Plain Text

Tab Width: 8

Ln 1, Col 1

INS

```
Open [icon] oooutput ~/Desktop Save [icon] - [icon] x
quicksort.cpp oooutput x
45 Call graph (explanation follows)
46
47
48 granularity: each sample hit covers 4 byte(s) for 0.07% of 13.77 seconds
49
50 index % time self children called name
51
52 [1] 100.0 0.00 13.77 1/1 main [1]
53 0.00 13.77 1/1 quickSort(int*, int, int) [3]
54 0.00 0.00 2/2 printArray(int*, int) [11]
55 -----
56 10.27 3.50 99999/99999 quickSort(int*, int, int) [3]
57 [2] 100.0 10.27 3.50 99999 partition(int*, int, int) [2]
58 3.50 0.00 2500049999/2500049999 swap(int*, int*) [4]
59 -----
60 199998 quickSort(int*, int, int) [3]
61 0.00 13.77 1/1 main [1]
62 [3] 100.0 0.00 13.77 1+199998 quickSort(int*, int, int) [3]
63 10.27 3.50 99999/99999 partition(int*, int, int) [2]
64 199998 quickSort(int*, int, int) [3]
65 -----
66 3.50 0.00 2500049999/2500049999 partition(int*, int, int) [2]
67 [4] 25.4 3.50 0.00 2500049999 swap(int*, int*) [4]
68 -----
69 0.00 0.00 2/2 main [1]
70 [11] 0.0 0.00 0.00 2 printArray(int*, int) [11]
71 -----
72 0.00 0.00 1/1 _GLOBAL_sub_I_Z4swapPiS_ [13]
73 [12] 0.0 0.00 0.00 1 __static_initialization_and_destruction_0(int, int) [12]
74 -----
75
76 This table describes the call tree of the program, and was sorted by
77 the total amount of time spent in each function and its children.
78
79 Each entry in this table consists of several lines. The line with the
```

```
Open [icon] oooutput ~/Desktop Save [icon] - [icon] x
quicksort.cpp oooutput x
1 Flat profile:
2
3 Each sample counts as 0.01 seconds.
4 % cumulative self self total
5 time seconds seconds calls s/call s/call name
6 74.58 10.27 10.27 99999 0.00 0.00 partition(int*, int, int)
7 25.42 13.77 3.50 2500049999 0.00 0.00 swap(int*, int*)
8 0.00 13.77 0.00 2 0.00 0.00 printArray(int*, int)
9 0.00 13.77 0.00 1 0.00 0.00 __static_initialization_and_destruction_0(int, int)
10 0.00 13.77 0.00 1 0.00 13.77 quickSort(int*, int, int)
11
12 % the percentage of the total running time of the
13 time program used by this function.
14
15 cumulative a running sum of the number of seconds accounted
16 seconds for by this function and those listed above it.
17
18 self the number of seconds accounted for by this
19 seconds function alone. This is the major sort for this
20 listing.
21
22 calls the number of times this function was invoked, if
23 this function is profiled, else blank.
24
25 self the average number of milliseconds spent in this
26 ms/call function per call, if this function is profiled,
27 else blank.
28
29 total the average number of milliseconds spent in this
30 ms/call function and its descendants per call, if this
31 function is profiled, else blank.
32
33 name the name of the function. This is the minor sort
34 for this listing. The index shows the location of
35 the function in the gprof listing. If the index is
```