Who Is to Blame?

coursera.org/learn/git-distributed-development/supplement/FYmTP/who-is-to-blame

It is possible to assign blame for whom is responsible for a given set of lines in a file. For example, using **file2** from the previous example:

```
$ git blame file2

f60c0c21 (A Smart Guy 2009-12-31 13:50:15 -0600 1) file2

4b4bf2c5 (A Smart Guy 2009-12-31 13:50:15 -0600 2) another line for file2
```

shows the responsible commit and author. It is possible to specify a range of lines and other parameters for the search. For example, for a more complicated file in the Linux kernel source (each line had to be broken because of width limitations here):

```
c7:/usr/src/linux>git blame -L 3107,3121 kernel/sched/core.c

e220d2dc kernel/sched.c      (Peter Zijlstra      2009-05-23 18:28:55 +0200 3107

 )

e418e1c2 kernel/sched.c      (Christoph Lameter   2006-12-10 02:20:23 -0800 3108

 ) #ifdef CONFIG_SMP

6eb57e0d kernel/sched.c      (Suresh Siddha       2011-10-03 15:09:01 -0700 3109

 )       rq->idle_balance = idle_cpu(cpu);

7caff66f kernel/sched/core.c (Daniel Lezcano      2014-01-06 12:34:38 +0100 3110

 )       trigger_load_balance(rq);

e418e1c2 kernel/sched.c      (Christoph Lameter   2006-12-10 02:20:23 -0800 3111

 ) #endif

265f22a9 kernel/sched/core.c (Frederic Weisbecker 2013-05-03 03:39:05 +0200 3112

 )       rq_last_tick_reset(rq);

^1da177e kernel/sched.c      (Linus Torvalds      2005-04-16 15:20:36 -0700 3113

 ) }

^1da177e kernel/sched.c      (Linus Torvalds      2005-04-16 15:20:36 -0700 3114
```

)

265f22a9 kernel/sched/core.c (Frederic Weisbecker 2013-05-03 03:39:05 +0200 3115

) #ifdef CONFIG_NO_HZ_FULL

265f22a9 kernel/sched/core.c (Frederic Weisbecker 2013-05-03 03:39:05 +0200 3116

) /**

265f22a9 kernel/sched/core.c (Frederic Weisbecker 2013-05-03 03:39:05 +0200 3117

) * scheduler_tick_max_deferment

265f22a9 kernel/sched/core.c (Frederic Weisbecker 2013-05-03 03:39:05 +0200 3118

) *

265f22a9 kernel/sched/core.c (Frederic Weisbecker 2013-05-03 03:39:05 +0200 3119

) * Keep at least one tick per second when

265f22a9 kernel/sched/core.c (Frederic Weisbecker 2013-05-03 03:39:05 +0200 3120

) * active task is running because the sch

265f22a9 kernel/sched/core.c (Frederic Weisbecker 2013-05-03 03:39:05 +0200 3121

) * yet completely support full dynticks