

# Debug Postgres with GDB

## Table Of Contents:

- About GDB
- How to attach GDB to postgres
- How to debug a code using GDB (With an example)
- Other options available in GDB

## About GDB:

GDB, the GNU Project debugger, allows you to see what is going on 'inside' another program while it executes, or what it was doing at the moment it crashed.

GDB can do four main things to help you catch bugs:

- Start your program, specifying anything that might affect its behavior.
- Make your program stop on specified conditions.
- Examine what has happened, when your program has stopped.
- Change things in your program, so you can experiment with correcting the effects of one bug and go on to learn about another.

## Attaching GDB to Postgres:

Open terminal and type

```
./bin/postgres -D data &  
./bin/psql test
```

Find the Process id of this client session by typing  
`test=# SELECT pg_backend_pid();`

This is the Process ID to which you need to attach the GDB and start debugging. The Snapshot is shown.

```
Terminal
sourabh@sourabh-HP-ENVY-m6-Notebook-PC: ~/tarballs/postgresql-9.3.5/pgsql$
sourabh@sourabh-HP-ENVY-m6-Notebook-PC:~/tarballs/postgresql-9.3.5/pgsql$
sourabh@sourabh-HP-ENVY-m6-Notebook-PC:~/tarballs/postgresql-9.3.5/pgsql$
/bin/psql test
psql (9.3.5)
Type "help" for help.

test=# select pg_backend_pid();
 pg_backend_pid
-----
          4620
(1 row)

test=#
```

```
sourabh@sourabh-HP-ENVY-m6-Notebook-PC:~$ sudo gdb
GNU gdb (Ubuntu 7.7-0ubuntu3.1) 7.7
Copyright (C) 2014 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word".
(gdb) attach 4620
```

GDB is opened in a separate terminal window. You are likely to need superuser permission to attach to the Postgres process if the UID's do not match.

To attach GDB to the process (PID=4620, in this case), type the following command

```
(gdb) attach 4620
```

Now that GDB is attached to the process, we can start the debugging process. Let us start by setting a breakpoint. In this example, I have set the breakpoint to line number 52 in the file hash.c. This function is

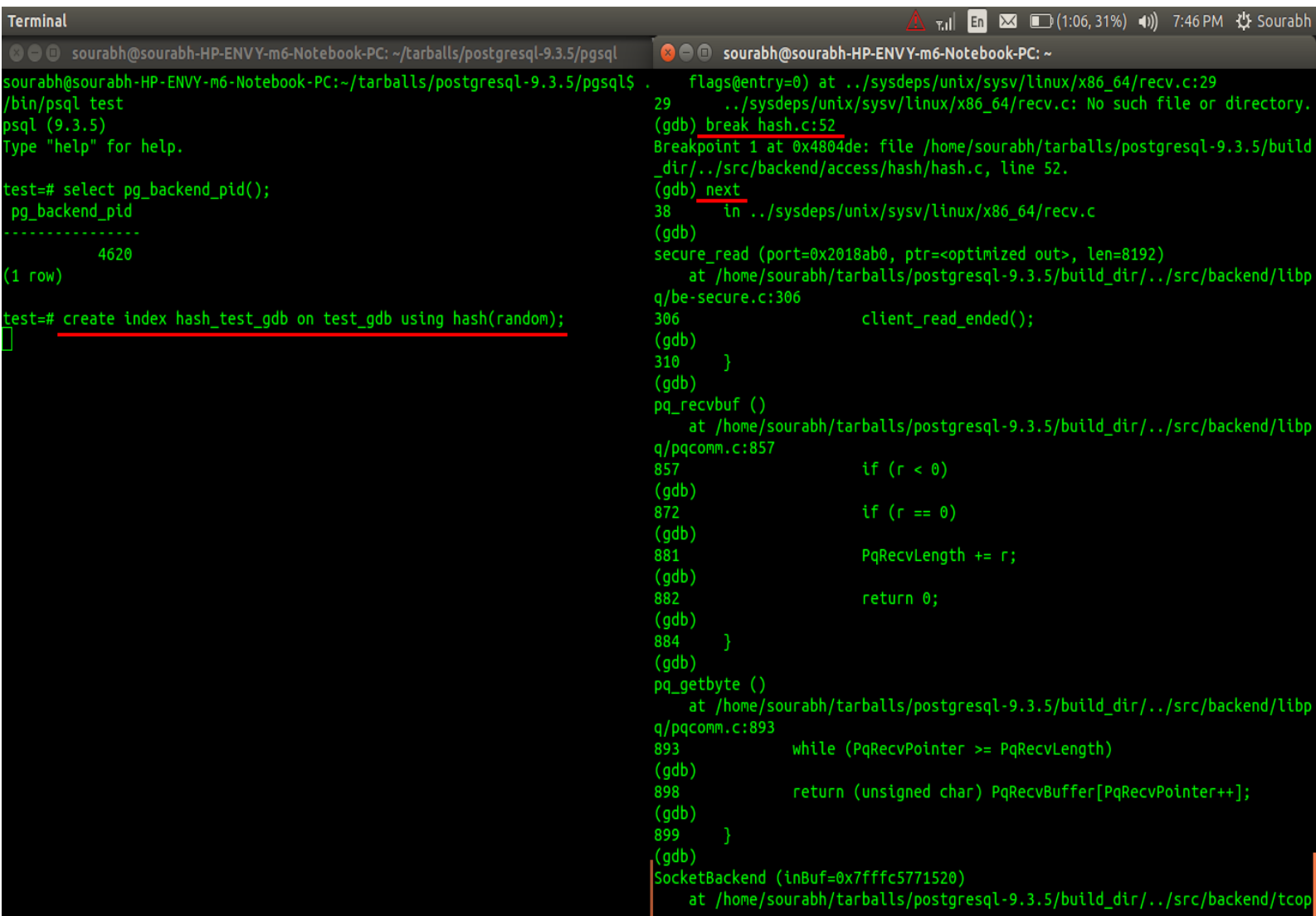
invoked when we create a hash index for a column.  
The way to set breakpoint is as follows

(gdb) break hash.c:52

Note:

1. You can also set it to break when it calls a particular function.
2. b can be used instead of break.
3. Any number of breakpoints can be set.

Now type the command to Create a hash index at the psql prompt and start debugging.  
Here's the screenshot.



```
Terminal
sourabh@sourabh-HP-ENVY-m6-Notebook-PC: ~/tarballs/postgresql-9.3.5/pgsql$
sourabh@sourabh-HP-ENVY-m6-Notebook-PC: ~/tarballs/postgresql-9.3.5/pgsql$ ./bin/psql test
psql (9.3.5)
Type "help" for help.

test=# select pg_backend_pid();
 pg_backend_pid
-----
          4620
(1 row)

test=# create index hash_test_gdb on test_gdb using hash(random);
[

(gdb) break hash.c:52
Breakpoint 1 at 0x4804de: file /home/sourabh/tarballs/postgresql-9.3.5/build_dir/./src/backend/access/hash/hash.c, line 52.
(gdb) next
38      in ./sysdeps/unix/sysv/linux/x86_64/recv.c
(gdb)
secure_read (port=0x2018ab0, ptr=<optimized out>, len=8192)
    at /home/sourabh/tarballs/postgresql-9.3.5/build_dir/./src/backend/libpq/be-secure.c:306
306
    client_read_ended();
(gdb)
310     }
(gdb)
pq_recvbuf ()
    at /home/sourabh/tarballs/postgresql-9.3.5/build_dir/./src/backend/libpq/pqcomm.c:857
857
    if (r < 0)
(gdb)
872
    if (r == 0)
(gdb)
881
    PqRecvLength += r;
(gdb)
882
    return 0;
(gdb)
884     }
(gdb)
pq_getbyte ()
    at /home/sourabh/tarballs/postgresql-9.3.5/build_dir/./src/backend/libpq/pqcomm.c:893
893
    while (PqRecvPointer >= PqRecvLength)
(gdb)
898
    return (unsigned char) PqRecvBuffer[PqRecvPointer++];
(gdb)
899     }
(gdb)
SocketBackend (inBuf=0x7ffc5771520)
    at /home/sourabh/tarballs/postgresql-9.3.5/build_dir/./src/backend/tcop
```

- The 'next' option allows us to check the next line of code that will be executed. 'n' can be used instead. This option jumps function calls
- The 'step' option (or 's') is used to enter into the function definition on function call
- Pressing 'Enter' repeats the last command

In the above screenshot, on typing next and pressing enter a few times, the codes related to accepting and processing the query are shown. Next we show the part where we reach the line number 52 in hash.c

```

Terminal
sourabh@sourabh-HP-ENVY-m6-Notebook-PC: ~/tarballs/postgresql-9.3.5/pgsql
sourabh@sourabh-HP-ENVY-m6-Notebook-PC: ~/tarballs/postgresql-9.3.5/pgsql$ ./bin/psql test
psql (9.3.5)
Type "help" for help.

test=# select pg_backend_pid();
 pg_backend_pid
-----
          4620
(1 row)

test=# create index hash_test_gdb on test_gdb using hash(random);

```

```

sourabh@sourabh-HP-ENVY-m6-Notebook-PC: ~
dbname=0x1ff9440 "test", username=<optimized out>
at /home/sourabh/tarballs/postgresql-9.3.5/build_dir/../src/backend/tcop/postgres.c:3979
3979         if (got_SIGHUP)
(gdb)
3973         DoingCommandRead = false;
(gdb)
3979         if (got_SIGHUP)
(gdb)
3989         if (ignore_till_sync && firstchar != EOF)
(gdb)
3992         switch (firstchar)
(gdb)
3999                                     SetCurrentStatementStartTime
stamp();
(gdb)
4001         query_string = pq_getmsgstri
ng(&input_message);
(gdb)
4002         pq_getmsgend(&input_message)
;
(gdb)
4001         query_string = pq_getmsgstri
ng(&input_message);
(gdb)
4002         pq_getmsgend(&input_message)
;
(gdb)
4004         if (am_walsender)
(gdb)
4007                                     exec_simple_query(qu
ery_string);
(gdb)

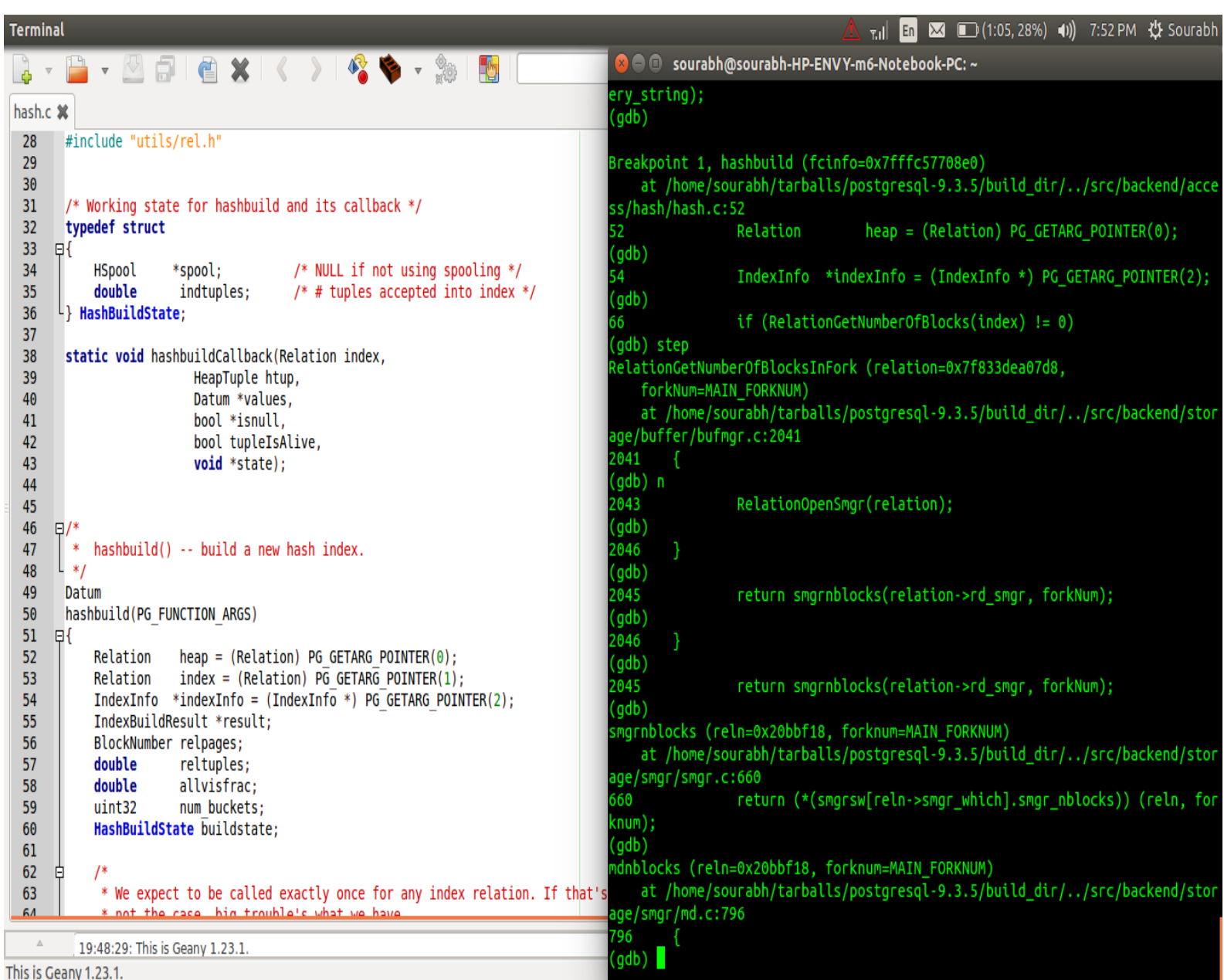
Breakpoint 1, hashbuild (fcinfo=0x7fffc57708e0)
at /home/sourabh/tarballs/postgresql-9.3.5/build_dir/../src/backend/acce
ss/hash/hash.c:52
52         Relation          heap = (Relation) PG_GETARG_POINTER(0);
(gdb)

```

At this point, We have reached out first (And in this case, the only) breakpoint. Keep pressing enter (next) till you reach the line

“if(RelationGetNumberOfBlocks(index) !=0)”

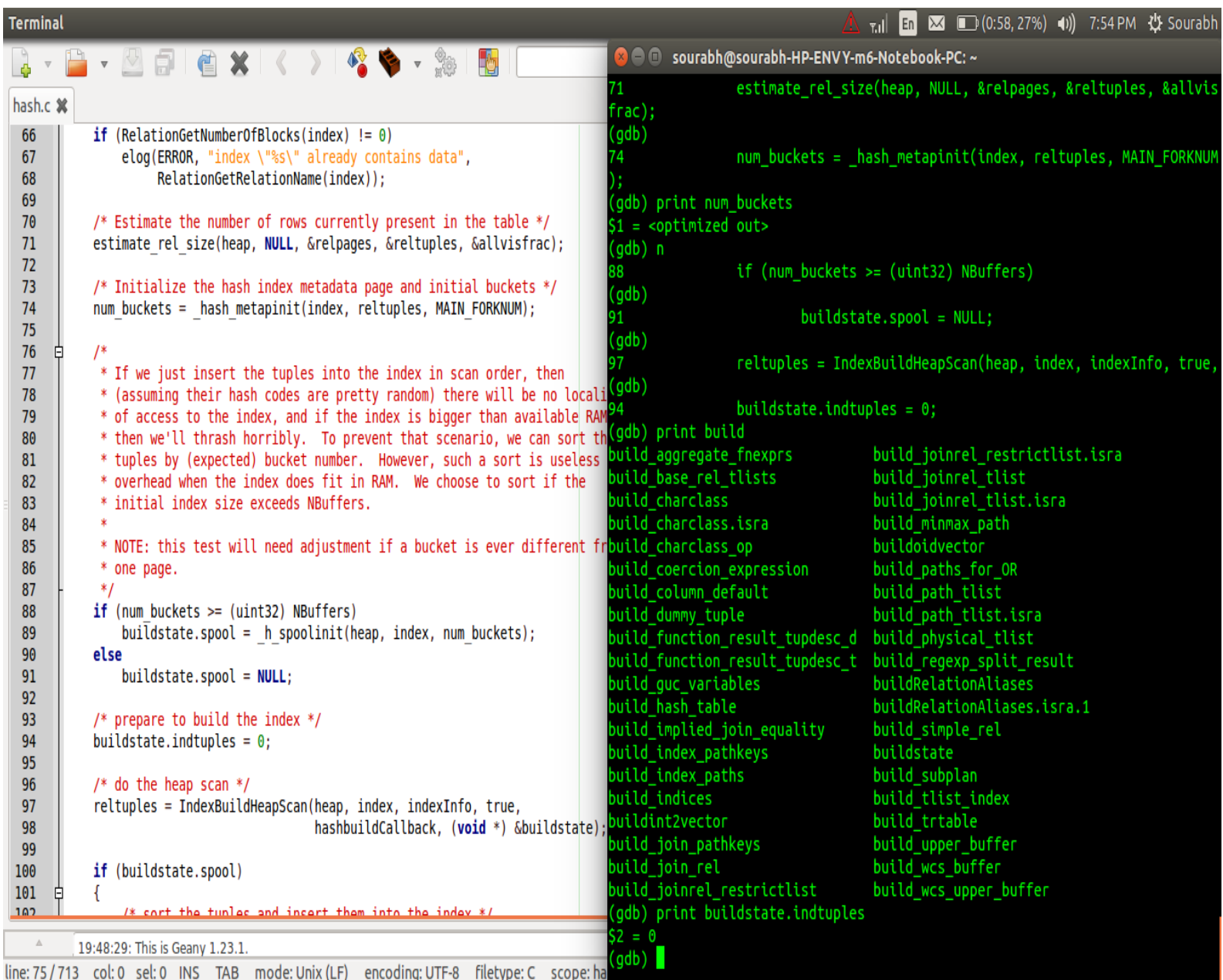
If you want to skip debugging this function call and go to the next line, type 'next'. Else type 'step'. Here we choose to go into the function definition by typing 'step'.



```
hash.c
28 #include "utils/rel.h"
29
30
31 /* Working state for hashbuild and its callback */
32 typedef struct
33 {
34     HSpool    *spool;        /* NULL if not using spooling */
35     double    indtuples;     /* # tuples accepted into index */
36 } HashBuildState;
37
38 static void hashbuildCallback(Relation index,
39                               HeapTuple htup,
40                               Datum *values,
41                               bool *isnull,
42                               bool tupleIsAlive,
43                               void *state);
44
45
46 /*
47  * hashbuild() -- build a new hash index.
48  */
49 Datum
50 hashbuild(PG_FUNCTION_ARGS)
51 {
52     Relation    heap = (Relation) PG_GETARG_POINTER(0);
53     Relation    index = (Relation) PG_GETARG_POINTER(1);
54     IndexInfo   *indexInfo = (IndexInfo *) PG_GETARG_POINTER(2);
55     IndexBuildResult *result;
56     BlockNumber relpages;
57     double      reltuples;
58     double      allvisfrac;
59     uint32      num_buckets;
60     HashBuildState buildstate;
61
62     /*
63      * We expect to be called exactly once for any index relation. If that's
64      * not the case, big trouble's what we have
65      */
66
67     every_string);
68     (gdb)
69
70     Breakpoint 1, hashbuild (fcinfo=0x7fffc57708e0)
71       at /home/sourabh/tarballs/postgresql-9.3.5/build_dir/../src/backend/acces
72       ss/hash/hash.c:52
73     52         Relation    heap = (Relation) PG_GETARG_POINTER(0);
74     (gdb)
75     54         IndexInfo   *indexInfo = (IndexInfo *) PG_GETARG_POINTER(2);
76     (gdb)
77     66         if (RelationGetNumberOfBlocks(index) != 0)
78     (gdb) step
79     RelationGetNumberOfBlocksInFork (relation=0x7f833dea07d8,
80       forkNum=MAIN_FORKNUM)
81       at /home/sourabh/tarballs/postgresql-9.3.5/build_dir/../src/backend/stor
82       age/buffer/bufmgr.c:2041
83     2041 {
84     (gdb) n
85     2043         RelationOpenSmgr(relation);
86     (gdb)
87     2046     }
88     (gdb)
89     2045         return smgrnblocks(relation->rd_smgr, forkNum);
90     (gdb)
91     2046     }
92     (gdb)
93     2045         return smgrnblocks(relation->rd_smgr, forkNum);
94     (gdb)
95     smgrnblocks (reln=0x20bbf18, forknum=MAIN_FORKNUM)
96       at /home/sourabh/tarballs/postgresql-9.3.5/build_dir/../src/backend/stor
97       age/smgr/smgr.c:660
98     660         return (*(smgrsw[reln->smgr_which].smgr_nblocks)) (reln, for
99       knum);
100     (gdb)
101     mdnblocks (reln=0x20bbf18, forknum=MAIN_FORKNUM)
102       at /home/sourabh/tarballs/postgresql-9.3.5/build_dir/../src/backend/stor
103       age/smgr/md.c:796
104     796 {
105     (gdb)
106     796 {
107     (gdb)
108     796 {
109     (gdb)
110     796 {
111     (gdb)
112     796 {
113     (gdb)
114     796 {
115     (gdb)
116     796 {
117     (gdb)
118     796 {
119     (gdb)
120     796 {
121     (gdb)
122     796 {
123     (gdb)
124     796 {
125     (gdb)
126     796 {
127     (gdb)
128     796 {
129     (gdb)
130     796 {
131     (gdb)
132     796 {
133     (gdb)
134     796 {
135     (gdb)
136     796 {
137     (gdb)
138     796 {
139     (gdb)
140     796 {
141     (gdb)
142     796 {
143     (gdb)
144     796 {
145     (gdb)
146     796 {
147     (gdb)
148     796 {
149     (gdb)
150     796 {
151     (gdb)
152     796 {
153     (gdb)
154     796 {
155     (gdb)
156     796 {
157     (gdb)
158     796 {
159     (gdb)
160     796 {
161     (gdb)
162     796 {
163     (gdb)
164     796 {
165     (gdb)
166     796 {
167     (gdb)
168     796 {
169     (gdb)
170     796 {
171     (gdb)
172     796 {
173     (gdb)
174     796 {
175     (gdb)
176     796 {
177     (gdb)
178     796 {
179     (gdb)
180     796 {
181     (gdb)
182     796 {
183     (gdb)
184     796 {
185     (gdb)
186     796 {
187     (gdb)
188     796 {
189     (gdb)
190     796 {
191     (gdb)
192     796 {
193     (gdb)
194     796 {
195     (gdb)
196     796 {
197     (gdb)
198     796 {
199     (gdb)
200     796 {
201     (gdb)
202     796 {
203     (gdb)
204     796 {
205     (gdb)
206     796 {
207     (gdb)
208     796 {
209     (gdb)
210     796 {
211     (gdb)
212     796 {
213     (gdb)
214     796 {
215     (gdb)
216     796 {
217     (gdb)
218     796 {
219     (gdb)
220     796 {
221     (gdb)
222     796 {
223     (gdb)
224     796 {
225     (gdb)
226     796 {
227     (gdb)
228     796 {
229     (gdb)
230     796 {
231     (gdb)
232     796 {
233     (gdb)
234     796 {
235     (gdb)
236     796 {
237     (gdb)
238     796 {
239     (gdb)
240     796 {
241     (gdb)
242     796 {
243     (gdb)
244     796 {
245     (gdb)
246     796 {
247     (gdb)
248     796 {
249     (gdb)
250     796 {
251     (gdb)
252     796 {
253     (gdb)
254     796 {
255     (gdb)
256     796 {
257     (gdb)
258     796 {
259     (gdb)
260     796 {
261     (gdb)
262     796 {
263     (gdb)
264     796 {
265     (gdb)
266     796 {
267     (gdb)
268     796 {
269     (gdb)
270     796 {
271     (gdb)
272     796 {
273     (gdb)
274     796 {
275     (gdb)
276     796 {
277     (gdb)
278     796 {
279     (gdb)
280     796 {
281     (gdb)
282     796 {
283     (gdb)
284     796 {
285     (gdb)
286     796 {
287     (gdb)
288     796 {
289     (gdb)
290     796 {
291     (gdb)
292     796 {
293     (gdb)
294     796 {
295     (gdb)
296     796 {
297     (gdb)
298     796 {
299     (gdb)
300     796 {
301     (gdb)
302     796 {
303     (gdb)
304     796 {
305     (gdb)
306     796 {
307     (gdb)
308     796 {
309     (gdb)
310     796 {
311     (gdb)
312     796 {
313     (gdb)
314     796 {
315     (gdb)
316     796 {
317     (gdb)
318     796 {
319     (gdb)
320     796 {
321     (gdb)
322     796 {
323     (gdb)
324     796 {
325     (gdb)
326     796 {
327     (gdb)
328     796 {
329     (gdb)
330     796 {
331     (gdb)
332     796 {
333     (gdb)
334     796 {
335     (gdb)
336     796 {
337     (gdb)
338     796 {
339     (gdb)
340     796 {
341     (gdb)
342     796 {
343     (gdb)
344     796 {
345     (gdb)
346     796 {
347     (gdb)
348     796 {
349     (gdb)
350     796 {
351     (gdb)
352     796 {
353     (gdb)
354     796 {
355     (gdb)
356     796 {
357     (gdb)
358     796 {
359     (gdb)
360     796 {
361     (gdb)
362     796 {
363     (gdb)
364     796 {
365     (gdb)
366     796 {
367     (gdb)
368     796 {
369     (gdb)
370     796 {
371     (gdb)
372     796 {
373     (gdb)
374     796 {
375     (gdb)
376     796 {
377     (gdb)
378     796 {
379     (gdb)
380     796 {
381     (gdb)
382     796 {
383     (gdb)
384     796 {
385     (gdb)
386     796 {
387     (gdb)
388     796 {
389     (gdb)
390     796 {
391     (gdb)
392     796 {
393     (gdb)
394     796 {
395     (gdb)
396     796 {
397     (gdb)
398     796 {
399     (gdb)
400     796 {
401     (gdb)
402     796 {
403     (gdb)
404     796 {
405     (gdb)
406     796 {
407     (gdb)
408     796 {
409     (gdb)
410     796 {
411     (gdb)
412     796 {
413     (gdb)
414     796 {
415     (gdb)
416     796 {
417     (gdb)
418     796 {
419     (gdb)
420     796 {
421     (gdb)
422     796 {
423     (gdb)
424     796 {
425     (gdb)
426     796 {
427     (gdb)
428     796 {
429     (gdb)
430     796 {
431     (gdb)
432     796 {
433     (gdb)
434     796 {
435     (gdb)
436     796 {
437     (gdb)
438     796 {
439     (gdb)
440     796 {
441     (gdb)
442     796 {
443     (gdb)
444     796 {
445     (gdb)
446     796 {
447     (gdb)
448     796 {
449     (gdb)
450     796 {
451     (gdb)
452     796 {
453     (gdb)
454     796 {
455     (gdb)
456     796 {
457     (gdb)
458     796 {
459     (gdb)
460     796 {
461     (gdb)
462     796 {
463     (gdb)
464     796 {
465     (gdb)
466     796 {
467     (gdb)
468     796 {
469     (gdb)
470     796 {
471     (gdb)
472     796 {
473     (gdb)
474     796 {
475     (gdb)
476     796 {
477     (gdb)
478     796 {
479     (gdb)
480     796 {
481     (gdb)
482     796 {
483     (gdb)
484     796 {
485     (gdb)
486     796 {
487     (gdb)
488     796 {
489     (gdb)
490     796 {
491     (gdb)
492     796 {
493     (gdb)
494     796 {
495     (gdb)
496     796 {
497     (gdb)
498     796 {
499     (gdb)
500     796 {
501     (gdb)
502     796 {
503     (gdb)
504     796 {
505     (gdb)
506     796 {
507     (gdb)
508     796 {
509     (gdb)
510     796 {
511     (gdb)
512     796 {
513     (gdb)
514     796 {
515     (gdb)
516     796 {
517     (gdb)
518     796 {
519     (gdb)
520     796 {
521     (gdb)
522     796 {
523     (gdb)
524     796 {
525     (gdb)
526     796 {
527     (gdb)
528     796 {
529     (gdb)
530     796 {
531     (gdb)
532     796 {
533     (gdb)
534     796 {
535     (gdb)
536     796 {
537     (gdb)
538     796 {
539     (gdb)
540     796 {
541     (gdb)
542     796 {
543     (gdb)
544     796 {
545     (gdb)
546     796 {
547     (gdb)
548     796 {
549     (gdb)
550     796 {
551     (gdb)
552     796 {
553     (gdb)
554     796 {
555     (gdb)
556     796 {
557     (gdb)
558     796 {
559     (gdb)
560     796 {
561     (gdb)
562     796 {
563     (gdb)
564     796 {
565     (gdb)
566     796 {
567     (gdb)
568     796 {
569     (gdb)
570     796 {
571     (gdb)
572     796 {
573     (gdb)
574     796 {
575     (gdb)
576     796 {
577     (gdb)
578     796 {
579     (gdb)
580     796 {
581     (gdb)
582     796 {
583     (gdb)
584     796 {
585     (gdb)
586     796 {
587     (gdb)
588     796 {
589     (gdb)
590     796 {
591     (gdb)
592     796 {
593     (gdb)
594     796 {
595     (gdb)
596     796 {
597     (gdb)
598     796 {
599     (gdb)
600     796 {
601     (gdb)
602     796 {
603     (gdb)
604     796 {
605     (gdb)
606     796 {
607     (gdb)
608     796 {
609     (gdb)
610     796 {
611     (gdb)
612     796 {
613     (gdb)
614     796 {
615     (gdb)
616     796 {
617     (gdb)
618     796 {
619     (gdb)
620     796 {
621     (gdb)
622     796 {
623     (gdb)
624     796 {
625     (gdb)
626     796 {
627     (gdb)
628     796 {
629     (gdb)
630     796 {
631     (gdb)
632     796 {
633     (gdb)
634     796 {
635     (gdb)
636     796 {
637     (gdb)
638     796 {
639     (gdb)
640     796 {
641     (gdb)
642     796 {
643     (gdb)
644     796 {
645     (gdb)
646     796 {
647     (gdb)
648     796 {
649     (gdb)
650     796 {
651     (gdb)
652     796 {
653     (gdb)
654     796 {
655     (gdb)
656     796 {
657     (gdb)
658     796 {
659     (gdb)
660     796 {
661     (gdb)
662     796 {
663     (gdb)
664     796 {
665     (gdb)
666     796 {
667     (gdb)
668     796 {
669     (gdb)
670     796 {
671     (gdb)
672     796 {
673     (gdb)
674     796 {
675     (gdb)
676     796 {
677     (gdb)
678     796 {
679     (gdb)
680     796 {
681     (gdb)
682     796 {
683     (gdb)
684     796 {
685     (gdb)
686     796 {
687     (gdb)
688     796 {
689     (gdb)
690     796 {
691     (gdb)
692     796 {
693     (gdb)
694     796 {
695     (gdb)
696     796 {
697     (gdb)
698     796 {
699     (gdb)
700     796 {
701     (gdb)
702     796 {
703     (gdb)
704     796 {
705     (gdb)
706     796 {
707     (gdb)
708     796 {
709     (gdb)
710     796 {
711     (gdb)
712     796 {
713     (gdb)
714     796 {
715     (gdb)
716     796 {
717     (gdb)
718     796 {
719     (gdb)
720     796 {
721     (gdb)
722     796 {
723     (gdb)
724     796 {
725     (gdb)
726     796 {
727     (gdb)
728     796 {
729     (gdb)
730     796 {
731     (gdb)
732     796 {
733     (gdb)
734     796 {
735     (gdb)
736     796 {
737     (gdb)
738     796 {
739     (gdb)
740     796 {
741     (gdb)
742     796 {
743     (gdb)
744     796 {
745     (gdb)
746     796 {
747     (gdb)
748     796 {
749     (gdb)
750     796 {
751     (gdb)
752     796 {
753     (gdb)
754     796 {
755     (gdb)
756     796 {
757     (gdb)
758     796 {
759     (gdb)
760     796 {
761     (gdb)
762     796 {
763     (gdb)
764     796 {
765     (gdb)
766     796 {
767     (gdb)
768     796 {
769     (gdb)
770     796 {
771     (gdb)
772     796 {
773     (gdb)
774     796 {
775     (gdb)
776     796 {
777     (gdb)
778     796 {
779     (gdb)
780     796 {
781     (gdb)
782     796 {
783     (gdb)
784     796 {
785     (gdb)
786     796 {
787     (gdb)
788     796 {
789     (gdb)
790     796 {
791     (gdb)
792     796 {
793     (gdb)
794     796 {
795     (gdb)
796     796 {
797     (gdb)
798     796 {
799     (gdb)
800     796 {
801     (gdb)
802     796 {
803     (gdb)
804     796 {
805     (gdb)
806     796 {
807     (gdb)
808     796 {
809     (gdb)
810     796 {
811     (gdb)
812     796 {
813     (gdb)
814     796 {
815     (gdb)
816     796 {
817     (gdb)
818     796 {
819     (gdb)
820     796 {
821     (gdb)
822     796 {
823     (gdb)
824     796 {
825     (gdb)
826     796 {
827     (gdb)
828     796 {
829     (gdb)
830     796 {
831     (gdb)
832     796 {
833     (gdb)
834     796 {
835     (gdb)
836     796 {
837     (gdb)
838     796 {
839     (gdb)
840     796 {
841     (gdb)
842     796 {
843     (gdb)
844     796 {
845     (gdb)
846     796 {
847     (gdb)
848     796 {
849     (gdb)
850     796 {
851     (gdb)
852     796 {
853     (gdb)
854     796 {
855     (gdb)
856     796 {
857     (gdb)
858     796 {
859     (gdb)
860     796 {
861     (gdb)
862     796 {
863     (gdb)
864     796 {
865     (gdb)
866     796 {
867     (gdb)
868     796 {
869     (gdb)
870     796 {
871     (gdb)
872     796 {
873     (gdb)
874     796 {
875     (gdb)
876     796 {
877     (gdb)
878     796 {
879     (gdb)
880     796 {
881     (gdb)
882     796 {
883     (gdb)
884     796 {
885     (gdb)
886     796 {
887     (gdb)
888     796 {
889     (gdb)
890     796 {
891     (gdb)
892     796 {
893     (gdb)
894     796 {
895     (gdb)
896     796 {
897     (gdb)
898     796 {
899     (gdb)
900     796 {
901     (gdb)
902     796 {
903     (gdb)
904     796 {
905     (gdb)
906     796 {
907     (gdb)
908     796 {
909     (gdb)
910     796 {
911     (gdb)
912     796 {
913     (gdb)
914     796 {
915     (gdb)
916     796 {
917     (gdb)
918     796 {
919     (gdb)
920     796 {
921     (gdb)
922     796 {
923     (gdb)
924     796 {
925     (gdb)
926     796 {
927     (gdb)
928     796 {
929     (gdb)
930     796 {
931     (gdb)
932     796 {
933     (gdb)
934     796 {
935     (gdb)
936     796 {
937     (gdb)
938     796 {
939     (gdb)
940     796 {
941     (gdb)
942     796 {
943     (gdb)
944     796 {
945     (gdb)
946     796 {
947     (gdb)
948     796 {
949     (gdb)
950     796 {
951     (gdb)
952     796 {
953     (gdb)
954     796 {
955     (gdb)
956     796 {
957     (gdb)
958     796 {
959     (gdb)
960     796 {
961     (gdb)
962     796 {
963     (gdb)
964     796 {
965     (gdb)
966     796 {
967     (gdb)
968     796 {
969     (gdb)
970     796 {
971     (gdb)
972     796 {
973     (gdb)
974     796 {
975     (gdb)
976     796 {
977     (gdb)
978     796 {
979     (gdb)
980     796 {
981     (gdb)
982     796 {
983     (gdb)
984     796 {
985     (gdb)
986     796 {
987     (gdb)
988     796 {
989     (gdb)
990     796 {
991     (gdb)
992     796 {
993     (gdb)
994     796 {
995     (gdb)
996     796 {
997     (gdb)
998     796 {
999     (gdb)
1000    796 {
1001    (gdb)
1002    796 {
1003    (gdb)
1004    796 {
1005    (gdb)
1006    796 {
1007    (gdb)
1008    796 {
1009    (gdb)
1010    796 {
1011    (gdb)
1012    796 {
1013    (gdb)
1014    796 {
1015    (gdb)
1016    796 {
1017    (gdb)
1018    796 {
1019    (gdb)
1020    796 {
1021    (gdb)
1022    796 {
1023    (gdb)
1024    796 {
1025    (gdb)
1026    796 {
1027    (gdb)
1028    796 {
1029    (gdb)
1030    796 {
1031    (gdb)
1032    796 {
1033    (gdb)
1034    796 {
1035    (gdb)
1036    796 {
1037    (gdb)
1038    796 {
1039    (gdb)
1040    796 {
1041    (gdb)
1042    796 {
1043    (gdb)
1044    796 {
1045    (gdb)
1046    796 {
1047    (gdb)
1048    796 {
1049    (gdb)
1050    796 {
1051    (gdb)
1052    796 {
1053    (gdb)
1054    796 {
1055    (gdb)
1056    796 {
1057    (gdb)
1058    796 {
1059    (gdb)
1060    796 {
1061    (gdb)
1062    796 {
1063    (gdb)
1064    796 {
1065    (gdb)
1066    796 {
1067    (gdb)
1068    796 {
1069    (gdb)
1070    796 {
1071    (gdb)
1072    796 {
1073    (gdb)
1074    796 {
1075    (gdb)
1076    796 {
1077    (gdb)
1078    796 {
1079    (gdb)
1080    796 {
1081    (gdb)
1082    796 {
1083    (gdb)
1084    796 {
1085    (gdb)
1086    796 {
1087    (gdb)
1088    796 {
1089    (gdb)
1090    796 {
1091    (gdb)
1092    796 {
1093    (gdb)
1094    796 {
1095    (gdb)
1096    796 {
1097    (gdb)
1098    796 {
1099    (gdb)
1100    796 {
1101    (gdb)
1102    796 {
1103    (gdb)
1104    796 {
1105    (gdb)
1106    796 {
1107    (gdb)
1108    796 {
1109    (gdb)
1110    796 {
1111    (gdb)
1112    796 {
1113    (gdb)
1114    796 {
1115    (gdb)
1116    796 {
1117    (gdb)
1118    796 {
1119    (gdb)
1120    796 {
1121    (gdb)
1122    796 {
1123    (gdb)
1124    796 {
1125    (gdb)
1126    796 {
1127    (gdb)
1128    796 {
1129    (gdb)
1130    796 {
1131    (gdb)
1132    796 {
1133    (gdb)
1134    796 {
1135    (gdb)
1136    796 {
1137    (gdb)
1138    796 {
1139    (gdb)
1140    796 {
1141    (gdb)
1142    796 {
1143    (gdb)
1144    796 {
1145    (gdb)
1146    796 {
1147    (gdb)
1148    796 {
1149    (gdb)
1150    796 {
1151    (gdb)
1152    796 {
1153    (gdb)
1154    796 {
1155    (gdb)
1156    796 {
1157    (gdb)
1158    796 {
1159    (gdb)
1160    796 {
1161    (gdb)
1162    796 {
1163    (gdb)
1164    796 {
1165    (gdb)
1166    796 {
1167    (gdb)
1168    796 {
1169    (gdb)
1170    796 {
1171    (gdb)
1172    796 {
1173    (gdb)
1174    796 {
1175    (gdb)
1176    796 {
1177    (gdb)
1178    796 {
1179    (gdb)
1180    796 {
1181    (gdb)
1182    796 {
1183    (gdb)
1184    796 {
1185    (gdb)
1186    796 {
1187    (gdb)
1188    796 {
1189    (gdb)
1190    796 {
1191    (gdb)
1192    796 {
1193    (gdb)
1194    796 {
1195    (gdb)
1196    796 {
1197    (gdb)
1198    796 {
1199    (gdb)
1200    796 {
1201    (gdb)
1202    796 {
1203    (gdb)
1204    796 {
1205    (gdb)
1206    796 {
1207    (gdb)
1208    796 {
1209    (gdb)
1210    796 {
1211    (gdb)
1212    796 {
1213    (gdb)
1214    796 {
1215    (gdb)
1216    796 {
1217    (gdb)
1218    796 {
1219    (gdb)
1220    796 {
1221    (gdb)
1222    796 {
1223    (gdb)
1224    796 {
1225    (gdb)
1226    796 {
1227    (gdb)
1228    796 {
1229    (gdb)
1230    796 {
1231    (gdb)
1232    796 {
1233    (gdb)
1234    796 {
1235    (gdb)
1236    796 {
1237    (gdb)
1238    796 {
1239    (gdb)
1240    796 {
1241    (gdb)
1242    796 {
1243    (gdb)
1244    796 {
1245    (gdb)
1246    796 {
1247    (gdb)
1248    796 {
1249    (gdb)
1250    796
```

What if you want to check the values of variables that are in the program? Type 'next' and reach upto line 94, where we can see an assignment statement: `buildstate.indtuples=0`

To see the value of the variable, type 'print' followed by the variable name as shown in the next screenshot.



The screenshot shows a terminal window with two panes. The left pane displays a C program named `hash.c` with line numbers 66 to 102. The right pane shows the execution of the program using GDB, with line numbers 71 to 94 visible. The GDB output shows the value of `buildstate.indtuples` as 0.

```
hash.c
66 if (RelationGetNumberOfBlocks(index) != 0)
67     elog(ERROR, "index \"%s\" already contains data",
68           RelationGetRelationName(index));
69
70 /* Estimate the number of rows currently present in the table */
71 estimate_rel_size(heap, NULL, &relpages, &reltuples, &allvisfrac);
72
73 /* Initialize the hash index metadata page and initial buckets */
74 num_buckets = _hash_metapinit(index, reltuples, MAIN_FORKNUM);
75
76 /*
77  * If we just insert the tuples into the index in scan order, then
78  * (assuming their hash codes are pretty random) there will be no locali
79  * of access to the index, and if the index is bigger than available RAM
80  * then we'll thrash horribly. To prevent that scenario, we can sort th
81  * tuples by (expected) bucket number. However, such a sort is useless
82  * overhead when the index does fit in RAM. We choose to sort if the
83  * initial index size exceeds NBuffers.
84  *
85  * NOTE: this test will need adjustment if a bucket is ever different fr
86  * one page.
87  */
88 if (num_buckets >= (uint32) NBuffers)
89     buildstate.spool = _h_spoolinit(heap, index, num_buckets);
90 else
91     buildstate.spool = NULL;
92
93 /* prepare to build the index */
94 buildstate.indtuples = 0;
95
96 /* do the heap scan */
97 reltuples = IndexBuildHeapScan(heap, index, indexInfo, true,
98                               hashbuildCallback, (void *) &buildstate);
99
100 if (buildstate.spool)
101 {
102     /* sort the tuples and insert them into the index */
103 }
```

```
sourabh@sourabh-HP-ENVY-m6-Notebook-PC: ~
71 estimate_rel_size(heap, NULL, &relpages, &reltuples, &allvis
frac);
(gdb)
74 num_buckets = _hash_metapinit(index, reltuples, MAIN_FORKNUM
);
(gdb) print num_buckets
$1 = <optimized out>
(gdb) n
88 if (num_buckets >= (uint32) NBuffers)
(gdb)
91 buildstate.spool = NULL;
(gdb)
97 reltuples = IndexBuildHeapScan(heap, index, indexInfo, true,
(gdb)
94 buildstate.indtuples = 0;
(gdb) print build
build_aggregate_fnexprs      build_joinrel_restrictlist.isra
build_base_rel_tlists        build_joinrel_tlist
build_charclass              build_joinrel_tlist.isra
build_charclass.isra         build_minmax_path
build_charclass.op           buildoidvector
build_coercion_expression    build_paths_for_OR
build_column_default         build_path_tlist
build_dummy_tuple            build_path_tlist.isra
build_function_result_tupdesc_d build_physical_tlist
build_function_result_tupdesc_t build_regexp_split_result
build_guc_variables          buildRelationAliases
build_hash_table             buildRelationAliases.isra.1
build_implied_join_equality  build_simple_rel
build_index_pathkeys         buildstate
build_index_paths            build_subplan
build_indices                build_tlist_index
buildint2vector              build_trtable
build_join_pathkeys          build_upper_buffer
build_join_rel               build_wcs_buffer
build_joinrel_restrictlist   build_wcs_upper_buffer
(gdb) print buildstate.indtuples
$2 = 0
(gdb)
```

19:48:29: This is Geany 1.23.1.  
line: 75 / 713 col: 0 sel: 0 INS TAB mode: Unix (LF) encoding: UTF-8 filetype: C scope: ha

Here are some other oprions that are available in GDB:

- help
  - List GDB Commands
- run
  - Start program execution from the beginning
- info breakpoints
  - List Breakpoints
- info threads
  - List threads in use
- info registers
  - List Registers in use
- until
  - Continue until you reach a particular line number
- where
  - Shows Line Number and Function you are in
- backtrace
  - Prints Stack backtrace
- delete
  - Deletes all breakpoints
- continue
  - Continue Execution to next breakpoint
- quit
  - Exit GDB Debugger



Usage of some of the commands are shown in the snapshots that follow:

```
Terminal
sourabh@sourabh-HP-ENVY-m6-Notebook-PC: ~/tarballs/postgresql-9.3.5/pgsql
sourabh@sourabh-HP-ENVY-m6-Notebook-PC: ~/tarballs/postgresql-9.3.5/pgsql$ /bin/psql test
psql (9.3.5)
Type "help" for help.

test=# select pg_backend_pid();
 pg_backend_pid
-----
          4620
(1 row)

test=# create index hash_test_gdb on test_gdb using hash(random);

```

```
buildint2vector          build_trtable
build_join_pathkeys      build_upper_buffer
build_join_rel           build_wcs_buffer
build_joinrel_restrictlist build_wcs_upper_buffer
(gdb) print buildstate.indtuples
$2 = 0
(gdb) help
List of classes of commands:

aliases -- Aliases of other commands
breakpoints -- Making program stop at certain points
data -- Examining data
files -- Specifying and examining files
internals -- Maintenance commands
obscure -- Obscure features
running -- Running the program
stack -- Examining the stack
status -- Status inquiries
support -- Support facilities
tracepoints -- Tracing of program execution without stopping the program
user-defined -- User-defined commands

Type "help" followed by a class name for a list of commands in that class.
Type "help all" for the list of all commands.
Type "help" followed by command name for full documentation.
Type "apropos word" to search for commands related to "word".
Command name abbreviations are allowed if unambiguous.
(gdb) info breakpoints
Num   Type       Disp Enb Address            What
1      breakpoint keep y  0x00000000004804de in hashbuild
                                           at /home/sourabh/tarballs
/postgresql-9.3.5/build_dir/./src/backend/access/hash/hash.c:52
      breakpoint already hit 1 time
(gdb) info threads
Id     Target Id         Frame
* 1    process 4620 "postgres" hashbuild (fcinfo=<optimized out>)
      at /home/sourabh/tarballs/postgresql-9.3.5/build_dir/./src/backend/acce
ss/hash/hash.c:94
(gdb) █
```

```
Terminal
sourabh@sourabh-HP-ENVY-m6-Notebook-PC: ~/tarballs/postgresql-9.3.5/pgsql$ ./postgresql-9.3.5/build_dir/./src/backend/access/hash/hash.c:52
/bin/psql test
psql (9.3.5)
Type "help" for help.

test=# select pg_backend_pid();
 pg_backend_pid
-----
          4620
(1 row)

test=# create index hash_test_gdb on test_gdb using hash(random);

CREATE INDEX
test=#
test=# \di
               List of relations
Schema |      Name      | Type | Owner  | Table
-----+-----+-----+-----+-----
public | btree_hash_index | index | sourabh | btree_test_table
public | btree_index      | index | sourabh | test_table
public | hash_index       | index | sourabh | test_table
public | hash_test_gdb    | index | sourabh | test_gdb
public | test_hash_index  | index | sourabh | hash_test_table
public | u_btree_hash_index | index | sourabh | u_btree_test_table
public | u_btree_index    | index | sourabh | test_table
(7 rows)

test=# \q
sourabh@sourabh-HP-ENVY-m6-Notebook-PC:~/tarballs/postgresql-9.3.5/pgsql$

(gdb) breakpoint already hit 1 time
(gdb) info threads
   Id Target Id         Frame
* 1   process 4620 "postgres" hashbuild (fcinfo=<optimized out>)
      at /home/sourabh/tarballs/postgresql-9.3.5/build_dir/./src/backend/acces
ss/hash/hash.c:94
(gdb) info registers
 rax      0x2          2
 rbx      0x7f833dea07d8 140201656190936
 rcx      0x1          1
 rdx      0x20dac98      34450584
 rsi      0x7f833dea07d8 140201656190936
 rdi      0x7f833de9c058 140201656172632
 rbp      0x7f833de9c058 0x7f833de9c058
 rsp      0x7fffc5770860 0x7fffc5770860
 r8        0x480450      4719696
 r9        0x7fffc5770880 140736506300544
 r10       0x7f833d7c8fe0 140201649016800
 r11       0x246         582
 r12       0x20dac98      34450584
 r13       0x20dac98      34450584
 r14       0x0           0
 r15       0x2           2
 rip      0x480549      0x480549 <hashbuild+121>
 eflags    0x283         [ CF SF IF ]
 cs        0x33         51
 ss        0x2b         43
 ds        0x0           0
 es        0x0           0
 fs        0x0           0
 gs        0x0           0
(gdb) delete
Delete all breakpoints? (y or n) y
(gdb) continue
Continuing.
[Inferior 1 (process 4620) exited normally]
(gdb) quit
sourabh@sourabh-HP-ENVY-m6-Notebook-PC:~$
```

Finally, when we type 'continue', the rest of the program gets executed since we have no more breakpoints and the index gets created.

Type Quit to exit the GDB Debugger.