Subject: Casting struct as void * segfault  
Posted by tfmeads on Sun, 20 Nov 2016 23:29:28 GMT

In my main code, I'm trying to insert vertex structs into my binheap struct. I first instantiate the binheap and a vertex struct, then cast it as void * and insert it into my binheap as such:

```c
void *v = (void *) vertex
insert(heap,v)
```

My insert function header is insert(binheap *, void *). As soon as I pass these variables in and run it, it segmentation faults on me without running even one line of the insert function.

Subject: Re: Casting struct as void * segfault  
Posted by tfmeads on Mon, 21 Nov 2016 01:31:17 GMT

Wow. Ok. I was under the impression you had to cast it explicitly as void *. You can just do this:

```c
void *v = vertex;
```

Yeah, it's been a long day.

Subject: Re: Casting struct as void * segfault  
Posted by Johnson on Mon, 21 Nov 2016 01:42:49 GMT

I highly doubt the segfault has anything to do with casting vertex to (void *). When you say it segfaults "without running even one line of the insert function", do you mean it doesn't call the insert function, or that it segfaults while running the first line of the insert function? Regardless, it's hard to help debug without seeing the code. All I can do is suggest stepping through the program with GDB or some other debugger so that, from within the insert function, you can do something like print v and print ((vertex *) v)->data just to make absolutely certain that the pointers are pointing to valid data.

As a side note, I don't think you even need to make the temporary variable v; I think you can just do insert(heap, vertex)

Subject: Re: Casting struct as void * segfault  
Posted by lusth on Mon, 21 Nov 2016 17:38:22 GMT

It is always legal to assign a data pointer to a void pointer without a cast, so insert(heap,vertex)
works.

The other direction needs a cast.