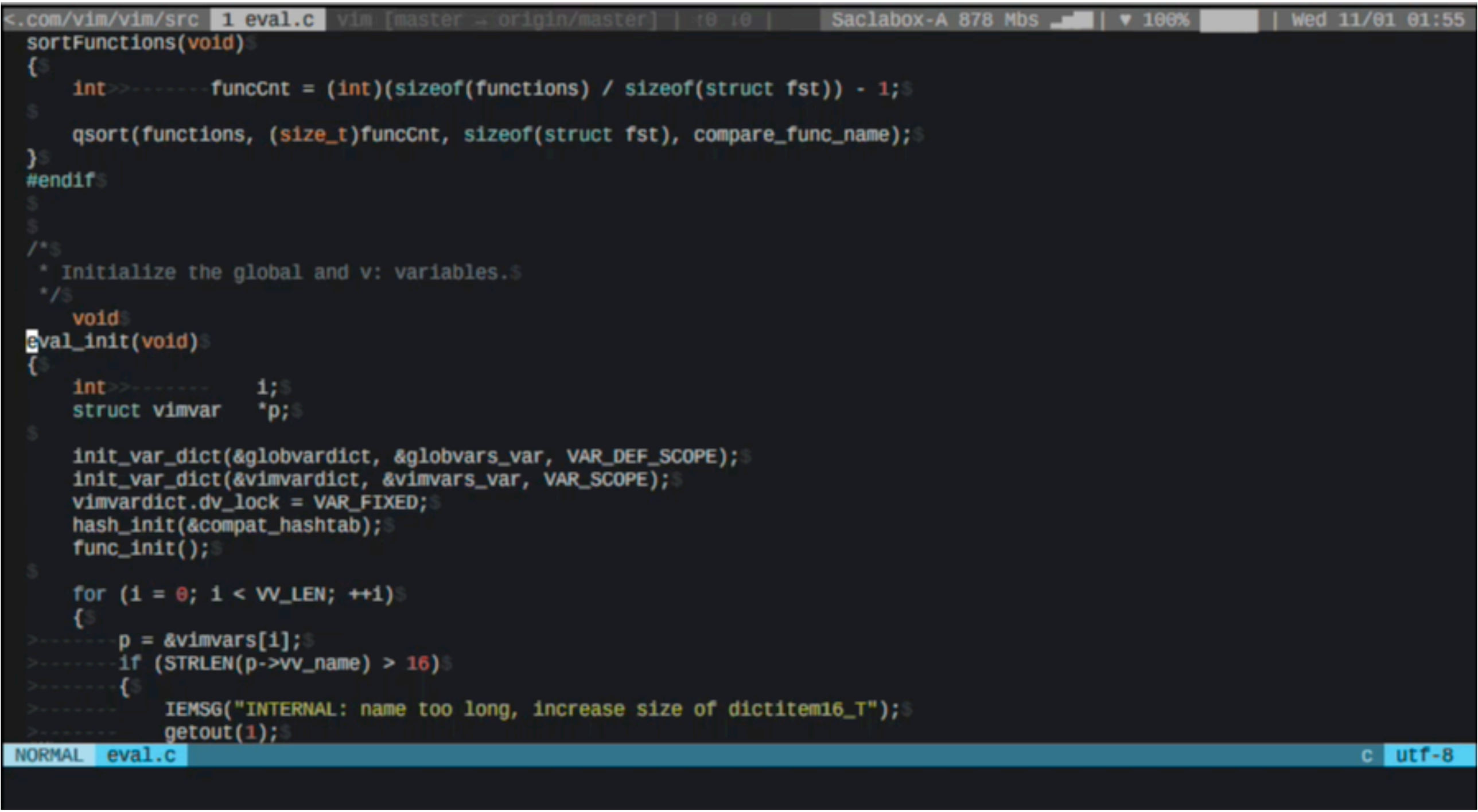
Blame on a large file

Focused to the part where you want to check.


~.com/vim/vim/src 1 eval.c vim [master → origin/master] | ↑0 ↓0 | Saclabox-A 878 Mbs | ▾ 100% | Wed 11/01 01:55
sortFunctions(void)
{
int funcCnt = (int)(sizeof(functions) / sizeof(struct fst)) - 1;
qsort(functions, (size_t)funcCnt, sizeof(struct fst), compare_func_name);
}
#endif
/*
* Initialize the global and v: variables.
*/
void
eval_init(void)
{
int i;
struct vimvar *p;
init_var_dict(&globvardict, &globvars_var, VAR_DEF_SCOPE);
init_var_dict(&vimvardict, &vimvars_var, VAR_SCOPE);
vimvardict.dv_lock = VAR_FIXED;
hash_init(&compat_hashtab);
func_init();
for (i = 0; i < VV_LEN; ++i)
{
p = &vimvars[i];
if (STRLEN(p->vv_name) > 16)
{
IEMSG("INTERNAL: name too long, increase size of dictitem16_T");
getout(1);
NORMAL eval.c c utf-8

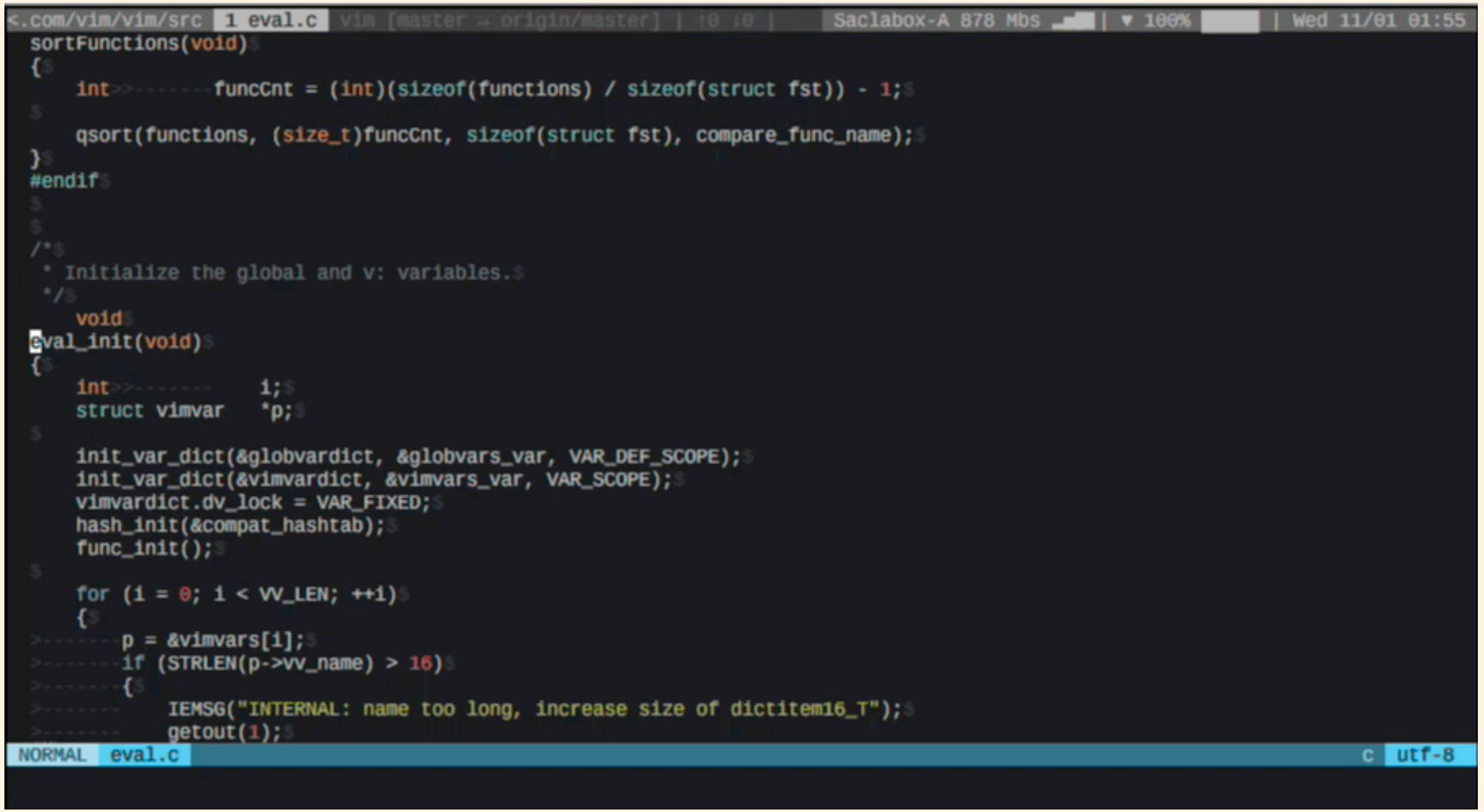
```
sortFunctions(void)
{
    int>>>>>>>>funcCnt = (int)(sizeof(functions) / sizeof(struct fst)) - 1;

    qsort(functions, (size_t)funcCnt, sizeof(struct fst), compare_func_name);
}
#endif


/*
 * Initialize the global and v: variables.
 */
    void
eval_init(void)
{
    int>>>>>>>>>>i;
    struct vimvar   *p;

    init_var_dict(&globvardict, &globvars_var, VAR_DEF_SCOPE);
    init_var_dict(&vimvardict, &vimvars_var, VAR_SCOPE);
    vimvardict.dv_lock = VAR_FIXED;
    hash_init(&compat_hashtab);
    func_init();

    for (i = 0; i < VV_LEN; ++i)
    {
>>>>>>>>p = &vimvars[i];
>>>>>>>>if (STRLEN(p->vv_name) > 16)
>>>>>>>>{
>>>>>>>>>>>>>IEMSG("INTERNAL: name too long, increase size of dictitem16_T");
>>>>>>>>>>>>>getout(1);
```



Blame on a large file

Focused to the part where you want to check.

Gina
Is
Namara
Awesome