

Define Interface

```
typedef struct fs<Type,Impl> {  
  __code ialloc(Impl* fs, uint dev, short type, __code next(...));  
} fs;
```

fs.dg

Implement

```
#interface "fs.dg"  
fs* createfs_impl(struct Context* cbc_context) {  
  fs->ialloc = C_iallocfs_impl;  
  return fs;  
}  
  
__code iallocfs_impl(struct fs_impl* fs, uint dev, short type, __code next(...)) {  
  goto allocinode(fs, dev, sb, next(...));  
}
```

fs_impl.cbc

Define implement header

```
typedef struct fs_impl<Impl, Isa> impl fs{  
  ...  
  __code allocinode(Type* fs_impl, uint dev, struct superblock* sb, __code  
  next(...));  
}
```

fs_impl.h

Implement

```
#interface "fs_impl.h"  
__code allocinode(struct fs_impl* fs_impl, uint dev, struct superblock* sb, __code next(...)){ //:skip  
  readsb(dev, sb);  
  Gearef(cbc_context, fs_impl)->inum = 1;  
  goto allocinode_loopcheck(fs_impl, inum, dev, sb, bp, dip, next(...));  
}
```

fs_impl_private.cbc

separate
implement