

```
template <class _Tp>
_LIBCPP_AVAILABILITY_ATOMIC_SHARED_PTR
void
atomic_store(shared_ptr<_Tp>* __p, shared_ptr<_Tp> __r)
{
    __sp_mut& __m = __get_sp_mut(__p);
    __m.lock();
    __p->swap(__r);
    __m.unlock();
}
```

`atomic<shared_ptr<T>>`

- `std::atomic<std::shared_ptr<T>>` probably won't be lock free :(