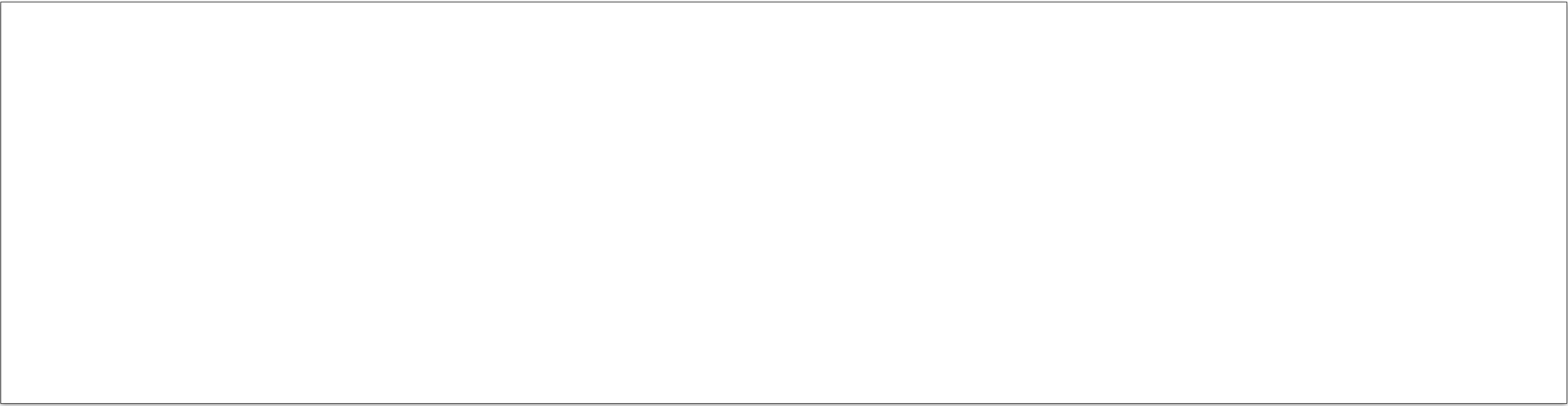






Data Races as Contract Violations

```
constexpr const_reference operator[](size_type __pos) const noexcept {  
    _LIBCPP_ASSERT_VALID_ELEMENT_ACCESS(__pos <= size(), "string index out of bounds");  
    scoped_check<check_type::read> _ (data_race_registry::get_state (this));  
  
    if (__builtin_constant_p(__pos) && !__fits_in_sso(__pos))  
        return *(__get_long_pointer() + __pos);  
  
    return *(data() + __pos);  
}
```







































































































































































W

































































W

































































































































































































































Data Races as Contract Violations

```
void read_started (check_state& state)
{
    ++state.num_readers;
    contract_assert (! state.is_writing); // read during active write
}

void write_started (check_state& state)
{
    contract_assert (! state.is_writing.exchange (true)) // write during active write
    contract_assert (state.num_readers == 0) // write during active read
}
```

```
constexpr const_reference operator[](size_type __pos) const noexcept {
    _LIBCPP_ASSERT_VALID_ELEMENT_ACCESS(__pos <= size(), "string index out of bounds");
    scoped_check<check_type::read> _ (data_race_registry::get_state (this));

    if (__builtin_constant_p(__pos) && !__fits_in_sso(__pos))
        return *(__get_long_pointer() + __pos);

    return *(data() + __pos);
}
```



Avoiding ABI Breaks: Extrinsic Storage

```
constexpr const_reference operator[](size_type __pos) const noexcept
    pre (can_read(data_race_registry::get_state (this)))
{
    //...
```

```
basic_string& replace(size_type __pos1, size_type __n1, const basic_string& __str)
    pre (can_write(data_race_registry::get_state (this)))
{
    //...
```