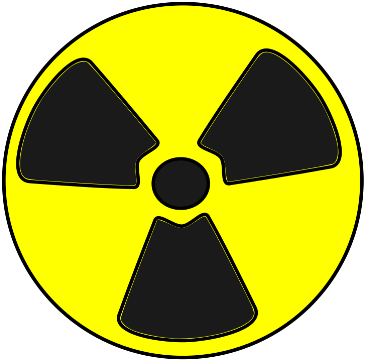
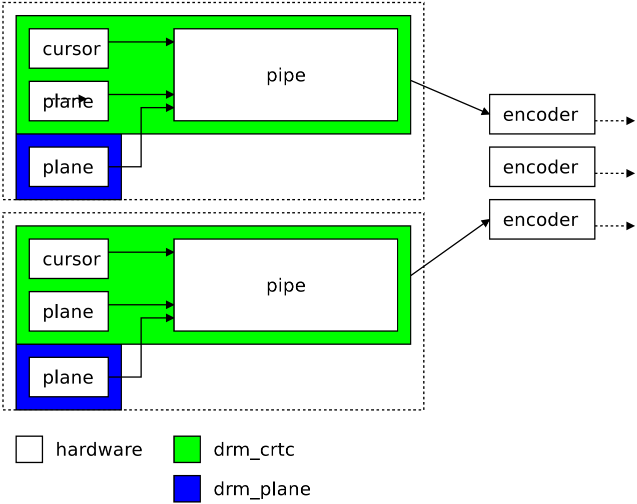
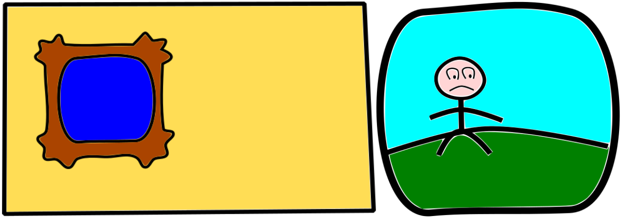
Atomic page flip and mode setting



Hardware structure and abstraction



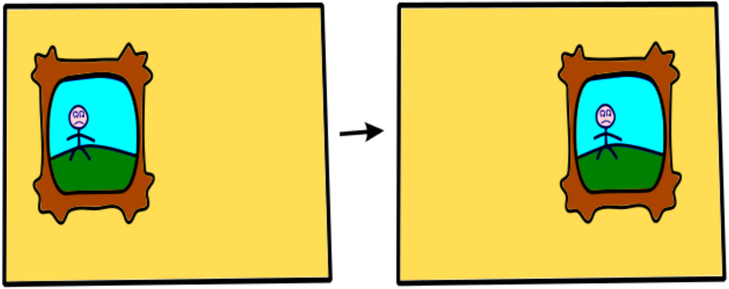
Atomic “page flip”



The hardware will compose the final image from two layers.

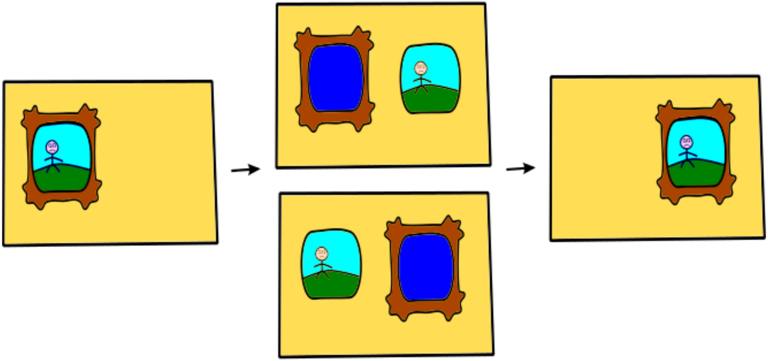
Atomic “page flip”

Animating the scene

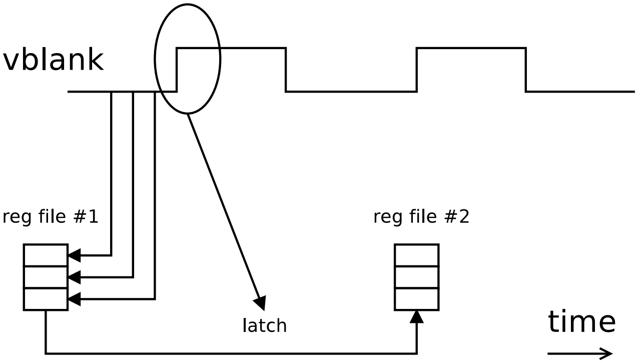


Atomic “page flip”

There are problems with animating the scene



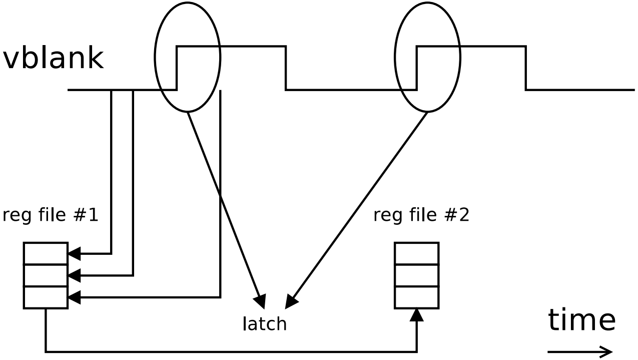
Double buffered registers



- Hardware maintains two register files
- First register file is written by the CPU
- Second register file is latched from the first one at vblank start

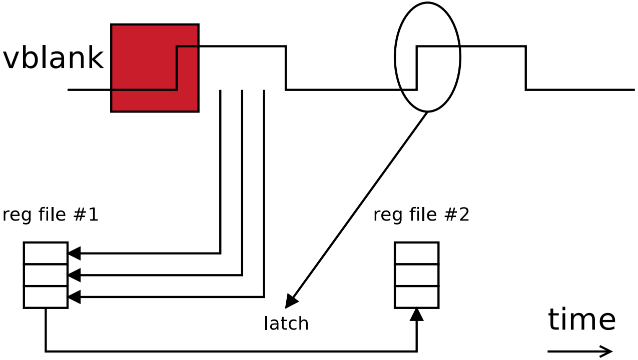
Double buffered registers

Problem: Hardware doesn't guarantee atomicity



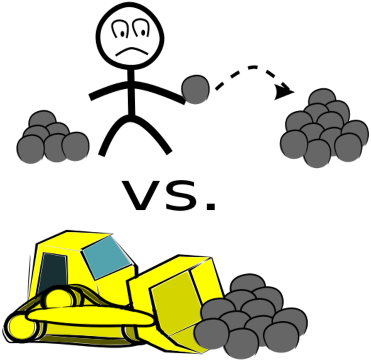
Double buffered registers

Solution: Prevent CPU from writing to registers near vblank start



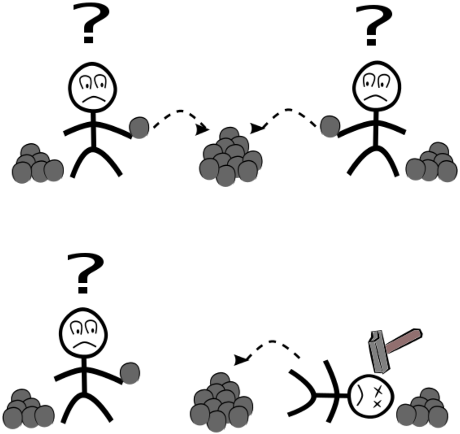
Kernel API

Transactional vs. one-shot

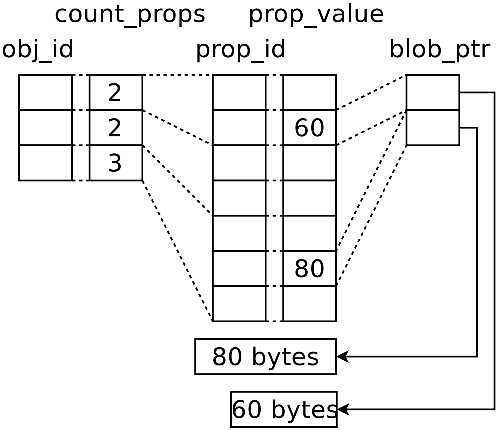


Kernel API

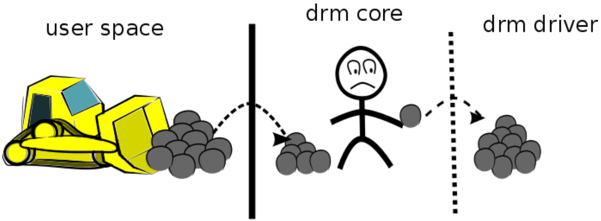
Possible issues with transactional API



Kernel API

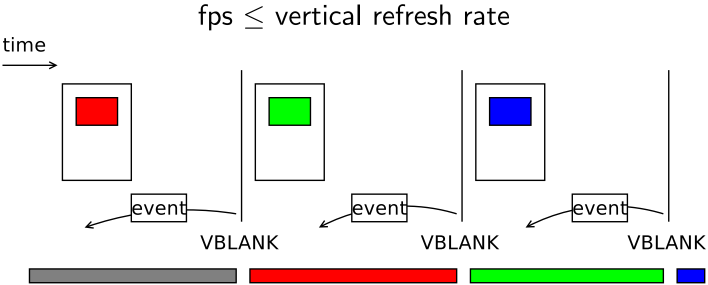

count_props
prop_value
obj_id
prop_id
blob_ptr
2
2
3
60
80
80 bytes
60 bytes

Kernel internal API



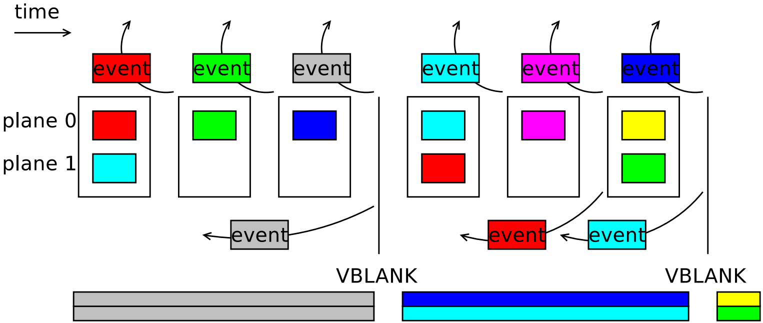
```
begin();
    set();
    set();
    set();
    ...
check();
commit();
end();
```

Completion events

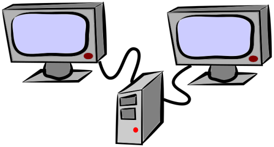


Completion events

fps > vertical refresh rate



Atomic mode setting



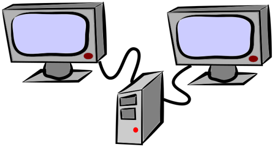
- Total shared resource available: 3N

Starting conditions:

- Display 1 resource requirement: N
- Display 2 resource requirement: N

$$N + N <= 3N$$

Atomic mode setting

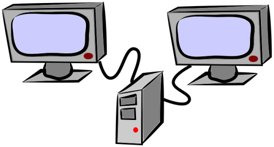


Step 1: Change mode on display 1

- Display 1 resource requirement: 2N
- Display 2 resource requirement: N

$$2N + N <= 3N$$

Atomic mode setting



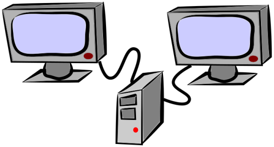
Step 2: Change mode on display 2

- Display 1 resource requirement: 2N
- Display 2 resource requirement: 2N

$$2N + 2N > 3N$$

Need to roll back mode change on display 1.

Atomic mode setting



Solution: Combine steps 1 and 2 into a single step.

The final state can be checked before the hardware state is clobbered, and thus there is no need for rolling back.