I/O Management and Disk Scheduling

Chapter 11

I/O Driver

- OS module which controls an I/O device
- hides the device specifics from the above layers in the OS/kernel
- translates logical I/O into device I/O (logical disk blocks into {track, head, sector})

- performs data buffering and scheduling of I/O operations
- structure: several synchronous entry points (device initialization, queue I/O requests, state control, read/write) and an asynchronous entry point (to handle interrupts)

Typical driver structure

```
driver_strategy(request)
{
    if (empty(request-queue))
        driver_start(request)
    else
        add(request, request-queue)
    block_current_process; reschedule()
}
driver_start(request) {
    current_request= request;
    start_dma(request);
}
driver_ioctl(request) {
}
```

```
driver_init() {
}
```

```
driver_interrupt(state) /* asynchronous part */
{
    if (state==ERROR) && (retries++<MAX) {
        driver_start(current_request);
        return;
    }
    add_current_process_to_active_queue
    if (! (empty(request_queue))
        driver_start(get_next(request_queue))
}
```