NOTES ON POSIX I/O

GRAYSON/WHITAKER

Questions.

1. In `sp_linux_copy.c`, why does `argv` have type `char*`, but we're indexing it by `1` and `2` to get `input_fd` and `output_fd`?
2. What exactly does the `[]` do in the function signature for `main`?
3. What does the following line do.

```
ret_out = write (output_fd, &buffer, (ssize_t) ret_in);
```

   Specifically, what does `(ssize_t) ret_in)` do? Is this a type cast? If so, what is the semantics of the type `ssize_t`?
4. Is `(EXIT_SUCCESS)` some kind of global constant? Is it defined by one of the `#include` statements?

Okay, so the obvious answer to the first question is that `argv` is an array of `char*` variables. Does the `[]` syntax do this for all types? It was mentioned that arrays and pointers are more or less the same thing because of pointer arithmetic in the last session. Does that apply here? What is going on in memory when we do something like `char * argv[];`? Can you even do that? Or is that syntax only valid in function signatures?

For question 3, we went over this a bit, but I'm confused about the difference between `ssize_t` and `size_t`. They're both essentially integers?