Lab 3 Extra-Credit

Publisher-subscriber systems like MQTT are usually used to send information when an event or change occurs. While many synchronous systems will set up a "stream" and continuously send small amounts of data, MQTT is designed to send a single block of data per update/event.

In the publisher/subscriber system you have set up, a publisher can only publish a single `uint32`. However, in most cases, a single integer is not enough to convey all of the information on an event. Furthermore, there is no standard format that publishers adhere to; some may want to send large amounts of data, while others transmit only small updates.

For extra credit, you will implement publishing a variable sized buffer of data in place of the `uint32` from parts (a) and (b). To accommodate this new requirement, the system calls will now have the following signatures:

- `syscall subscribe(topic16 t, void (*hnd)(topic16, void*, uint32))`
- `syscall unsubscribe(topic16 t)`
- `syscall publish(topic16 t, void* data, uint32 size)`

Requirements

The system calls will work in the same manner as the previous parts; the only difference is processes can now receive an array of data rather than a single integer. The full array of data should be available to the subscriber until the handler function returns.

Implementation Details

Availability is very important in this assignment. Once data has been published, subscribers should receive it in that state even if the publisher changes it's copy before a handler is called. Consider the following situation:

```
char data[5] = { 1, 2, 3, 4, 5 };
publish(0x2, data, 5);
data[2] = 0;
```

Imagine a subscriber has its handler called after this code excerpt. The data passed to the handler should be `{ 1, 2, 3, 4, 5}`, **not** `{ 1, 2, 0, 4, 5 }`. The kernel must ensure the original data is maintained until all handlers for that topic execute.

Submitting

Before beginning on the extra credit, create a copy of your completed Lab 3. When submitting, you should have two .zip files: one of all the files needed for the original Lab 3 (supporting the original single `uint32`), and one for the Lab 3 Extra-Credit.