Assessment 2 component: Destination select controller

Summary

Each instance of this controller is responsible for a destination select panel, and one will exist for each floor that an elevator can access. Employees swipe their cards, then press a button to select a floor. Destination select controllers are almost identical to card reader controllers, except with the additional step of reading a floor press. The overseer is responsible for checking floor authorisations.

Program name

```
destselect
```

Command-line arguments

```
{id} {wait time (in microseconds)} {shared memory path} {shared memory offset} {overseer address:port}
```

Shared memory structure

```
struct {
    char scanned[16];
    uint8_t floor_select;
    pthread_mutex_t mutex;
    pthread_cond_t scanned_cond;

    char response; // 'Y' or 'N' (or '\0' at first)
    pthread_cond_t response_cond;
};
```

Initialisation

On startup, this component will send the following initialisation message to the overseer via TCP:

`DESTSELECT {id} HELLO#`

Normal operation

After initialisation, the component will perform the following loop:

1. Lock the mutex
2. Look at the scanned code. If the bytes are all `\0` (NUL) nothing has been scanned yet- skip to 8
3. Open a TCP connection to the overseer
4. Send the following data: `DESTSELECT {id} SCANNED {scanned} {floor}#`
5. Wait up to {wait time} microseconds for a response from the overseer
6. If the response was `ALLOWED#`, set the 'response' char to `Y`. Anything else, or if the overseer timed out or the connection failed, set the response char to `N`
7. Signal 'response_cond'
8. Wait on 'scanned_cond'
9. Loop back to 2

Example operation

The program might be executed from the command-line with the following:

```
./destselect 715 /shm 1480 127.0.0.1:3000
```

The program will shm_open shared memory segment at /shm with an offset of 1480 and the size of the struct defined above, and mmap it into memory.

It will then open up a TCP connection to 127.0.0.1 on port 3000 and send the following text:

`DESTSELECT 715 HELLO#`

It will then immediately close the connection, and then proceed to the 'Normal operation' stage. If, for example, the 'scanned' array contains the following bytes: `59447a09ffc1c9c5` and the 'floor select' value contains the value 4, it will send this TCP message to 127.0.0.1:3000:

`DESTSELECT 715 SCANNED 59447a09ffc1c9c5 4#`

Then, depending on the response received, it will either write a Y or N into the 'response' char of the shared memory structure.