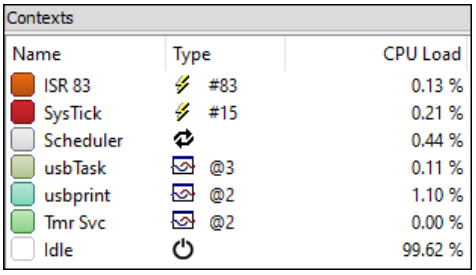
- My test, using this modified code:

```
txStream = xStreamBufferCreate( txBuffLen, 500);
uint8_t numBytes = xStreamBufferReceive(txStream, usbTxBuff, txBuffLen,
portMAX_DELAY);
```



4 Section: Using mutexes for access control (pg 308)

- **Clarification**, page 309
  - This paragraph was confusing to me:
    - *To prevent multiple copies of this mutex from being created for each compilation unit VirtualComDriverMultitTask is included in, we won't define our private global variables as having static scope this time. Since we don't have namespaces in C, we'll prepend the names with vcom_ in an attempt to avoid naming collisions with other globals.*
  - The paragraph is describing `VirtualCommDriverMultiTask.c`, and its private global variables. The paragraph says those variables are defined in a way that is amenable for use in multiple compilation units. Apparently, those variables are:

```
uint8_t vcom_usbTxBuff[txBuffLen];
StreamBufferHandle_t vcom_rxStream = NULL;
StreamBufferHandle_t vcom_txStream = NULL;
TaskHandle_t vcom_usbTaskHandle = NULL;
SemaphoreHandle_t vcom_mutexPtr = NULL;
```

  - In the paragraph, a contrast is made with `VirtualCommDriver.c`. That file's private global variables are defined as static. Apparently, those variables are:

```
static uint8_t usbTxBuff[txBuffLen];
static StreamBufferHandle_t txStream = NULL;
static TaskHandle_t usbTaskHandle = NULL;
```

  - The paragraph is confusing because support for multiple compilation units isn't actually implemented in `VirtualCommDriverMultiTask.c`.
  - For example, the private global variables would need to be defined in a header file. There, a conditional macro is needed to support the `extern` prefix. This use of header files is described here:
    - http://www.mathcs.emory.edu/~cheung/Courses/255/Syllabus/1-C-intro/cprep2.html