

## Tutorial: Wait and Signal

- copy the `simplesync.py` from the last lecture and run it
- copy the `simplemutex.py` from the last lecture and run it
- why does the output become scrambled?
- in the `simplemutex.py` program change all `print` statements into `printf` statements using appropriate parameters

## Tutorial: Wait and Signal

- use the following implementation of `printf` in Python

```
def printf (format, *args):  
    print str(format) % args,
```

- now add another mutex semaphore to protect the `print` statement inside `printf`
- does the output from the program become easier to read?

## Tutorial: Wait and Signal

- finally if you have time consider how you might solve the readers and writers problem using semaphores
- the readers and writers problem is defined as follows
  - only one writer can enter a section of code at a time
  - if no writer is in the section of code then any number of readers can enter at a time
  - a writer cannot enter until all readers have exited the section of code