https://github.com/klynch416/BIOL432_A9.git (https://github.com/klynch416/BIOL432_A9.git)

Libraries

```
library(doParallel)
```

```
## Loading required package: foreach
```

```
## Loading required package: iterators
```

```
## Loading required package: parallel
```

```
library(parallel)
library(foreach)
```

Number of cores in my system.

```
detectCores()
```

```
## [1] 16
```

Loop Write a standard for loop 4,000 times, and each time calculate the mean of 100,000 random numbers from a normal distribution with mean of 10 and standard deviation of 3.

```
before <- Sys.time()

table <- matrix("NA", nrow = 4000)

series <- for (i in 1:4000){
  i <- rnorm(100000, mean = 10, sd = 3)
}

time <- Sys.time() - before
print(time)
```

```
## Time difference of 17.12043 secs
```

Calculated time with multiple processors

```
multtime <- (as.numeric(gsub("(\\n)", "\\1", time))/detectCores())
print(multtime)
```

```
## [1] 1.070027
```

Loop with multiple threads

```
Cores <- makeCluster(detectCores())
registerDoParallel(Cores)

beforemult <- Sys.time()

paralle <- foreach(i = 4000, .combine = cbind) %dopar% {
  i <- rnorm(100000, mean = 10, sd = 3)
}

stopCluster(Cores)

print(Sys.time() - beforemult)
```

```
## Time difference of 0.0496769 secs
```

The serial loop took about 17 seconds to complete whereas the parallel program took less than a second to complete. The theoretical time was just slightly longer than the actual run-time. This could be due to the theoretical time being calculated off one core being used but when multiple are used they enhance the reliability of the others, making the process quicker.