

进程和线程实验报告

物理试验班 51 李昂 2150201074

实验原理

通过编写多进程的并发服务器-客户端程序和多线程的并发服务器-客户端程序来比较服务器端资源（主要是内存）的开销，进而体现线程和进程在资源使用上的不同。

实验平台：Macos Catalina 10.15

代码实现

编写服务器端和客户端程序，运用socket网络编程实现本机端口间的通信。当连接建立后，服务器分别创建新的进程和线程来处理客户端的请求，此处为简单的发送日期和时间。代码见附件

客户端 tcp_client.cpp

服务器端（多进程并发） tcp_server_process.cpp

服务器端（多线程并发） tcp_server_thread.cpp

实验过程

执行编译

```
clang++ -o tcp_server_process tcp_server_process.cpp
```

```
clang++ -o tcp_server_thread -lpthread tcp_server_thread.cpp
```

```
//由于使用了pthread库所以编译时要加上-lpthread.
```

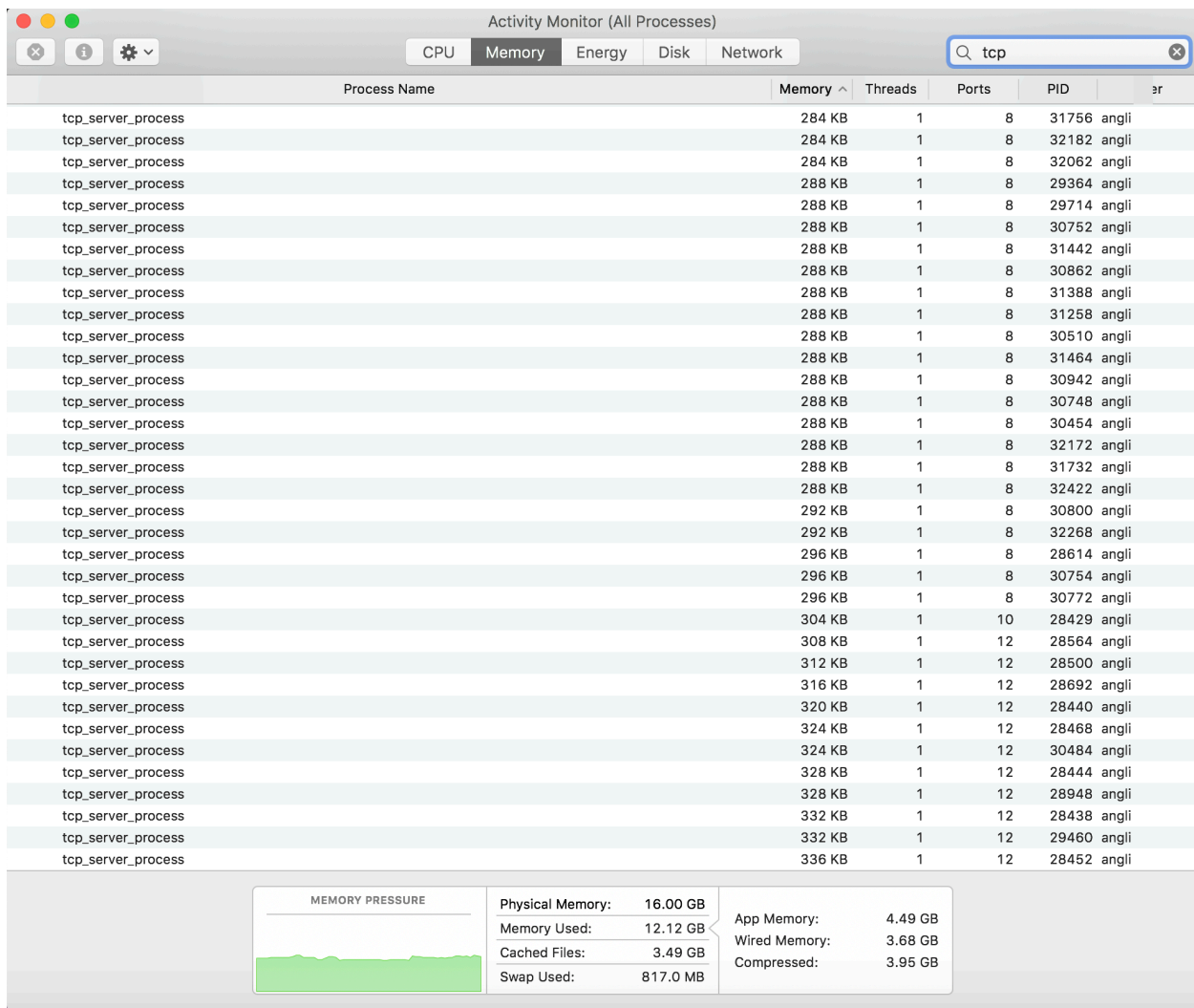
```
clang++ -o tcp_client tcp_client.cpp
```

1.使用多进程并发服务器端

```
./tcp_server_process 12321 #服务器端监听本地127.0.0.1的12321端口
```

```
./tcp_client 127.0.0.1 12321
```

```
#客户端向本地127.0.0.1的12321端口发出请求，通过运行client.sh 让客户端请求2000次
```

每个进程的线程数为1，每个进程所占用的内存资源在240KB到340KB之间，平均为每个进程约为300KB，而这样的进程有2000个，共占用约600MB内存资源。

2.使用多线程并发服务器端

`./tcp_server_thread 12321` #服务器端监听本地127.0.0.1的12321端口

`./tcp_client 127.0.0.1 12321`

#客户端向本地127.0.0.1的12321端口发出请求，通过运行client.sh 让客户端请求2000次

实验结论

通过这两组实验对比，同样是客户端发出2000次请求，多进程并发客户端消耗了约600MB的内存资源，而多线程并发客户端只消耗了16.1MB的内存资源。可见通过多线程并发能显著降低对内存资源的消耗。

参考资料：

<https://www.cnblogs.com/SeekHit/p/6537932.html>

Stack Overflow

www.geeksforgeeks.org

CSDN

《Operating.System.Concepts.10th.Edition》

本实验 GitHub:

<https://github.com/ang830715/processvsthread.git>