

Лекция №0

# Многопоточное программирование

Дмитрий Калугин-Балашов

# Преподаватели

---



**Дмитрий Калугин-Балашов**

Руководитель группы разработки

**Виктор Стародуб**

Программист



Работа с памятью (вводная лекция)

Работа с сетью (2 лекции)

Работа с процессами (2 лекции)

Работа с потоками (1 лекция)

Параллельное программирование (1 лекция)



## **Работа с памятью (вводная лекция)**

Работа с сетью (2 лекции)

Работа с процессами (2 лекции)

Работа с потоками (1 лекция)

Параллельное программирование (1 лекция)



Работа с памятью (вводная лекция)

**Работа с сетью (2 лекции)**

Работа с процессами (2 лекции)

Работа с потоками (1 лекция)

Параллельное программирование (1 лекция)



Работа с памятью (вводная лекция)

Работа с сетью (2 лекции)

**Работа с процессами (2 лекции)**

Работа с потоками (1 лекция)

Параллельное программирование (1 лекция)



Работа с памятью (вводная лекция)

Работа с сетью (2 лекции)

Работа с процессами (2 лекции)

**Работа с потоками (1 лекция)**

Параллельное программирование (1 лекция)



Работа с памятью (вводная лекция)

Работа с сетью (2 лекции)

Работа с процессами (2 лекции)

Работа с потоками (1 лекция)

**Параллельное программирование (1 лекция)**



# Практические занятия

---



**После каждой лекции (и по теме лекции).**

**Количество баллов за каждое занятие – 10.**

# Домашнее задание



## Проект «Маленькой nginx»

Пишем web-сервер.

Две контрольные точки – промежуточная (коллоквиум) и финальная.

В промежуточную сдаем **рабочий** веб-сервер без многопоточности.

К финальной сдаем многопоточность.

По **15 баллов** за этап.



«5» - 100

«4» - 80

«3» - 60

# Тест





`std::list<int>`



```
std::map<int, int>
```



`std::forward_list<int>`



---

```
std::unordered_map<int, int>
```





`boost::circular_buffer<int>`



`std::auto_ptr<Class>`



---

```
std::unique_ptr<Class>
```



`std::shared_ptr<Class>`



`std::weak_ptr<Class>`



---

malloc(1024)



`calloc(128, 16)`



---

```
realloc(pointer, 2048)
```





`free(ptr)`



fork()



---

`wait(&status)`



```
waitpid(-1, &status, 0)
```



`kill(pid, SIGUSR1)`



```
shmget(key, SHMSZ,  
IPC_CREAT | 666)
```



```
socket(AF_INET,  
SOCK_STREAM,  
IPPROTO_TCP)
```



```
pthread_create(&thread,  
NULL, thread_func, NULL)
```





---

pthread\_mutex\_t



---

pthread\_rwlock\_t



---

#pragma omp parallel



```
tbb::parallel_for<size_t>(1,  
    10, 1, print)
```



```
tbb::parallel_for<size_t>(1,  
    10, 1, print)
```



```
g++ -o daemon daemon.cpp -lz
```



(gdb) bt



(gdb) thr 22





(gdb) make -j 8



`strace -p 1022`



lsof -p 1102

Тест



---

htop



```
LD_PRELOAD=/usr/local/libaaa.so
```



ldd ./bin/daemon



ldconfig -p



---

`ulimit -c unlimited`





**Спасибо за  
внимание!**

**Дмитрий Калугин-Балашов**

[rvncerr@rvncerr.org](mailto:rvncerr@rvncerr.org)