Wątki

Wątki

Główny wątek:

```
int main() {

//stwórz wątek  --------------------------->       void func(){

                                                   //wątek

//czekaj na wątek      <-----------------------   }

return 0;

}
```

Wątki

```
#include <thread>

void threadFn(int x){

cout<<”threadFn”<<endl;

}

int main(){

int x;

thread t1(threadFn,x);

t1.join();

return 0;

}
```

Wątki

```
#include <thread>

void threadFn(int & x){ ←---------------------

cout<<”threadFn”<<endl;

}

int main(){

int x;

thread t1(threadFn,ref(x));  ←-------------------------

t1.join();

return 0;

}
```

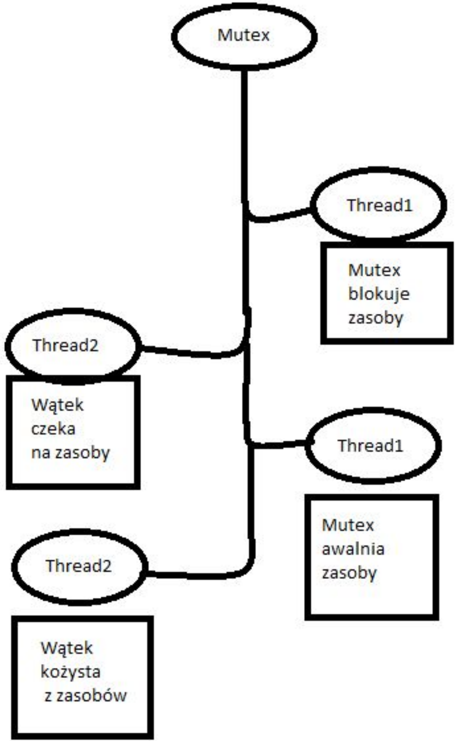
Wątki

```
#include <thread>


int main(){
int x;
thread t1([](int & tx)){
cout<<tx<<endl;
},ref(x);
t1.join();
return 0;
}
```

Wątki

```
#include <thread>


int main(){
int x;
thread t1([&](){
cout<<x<<endl;
});
t1.join();
return 0;
}
```

Mutex


Mutex
Thread1
Mutex blokuje zasoby
Thread2
Wątek czeka na zasoby
Thread1
Mutex awalnia zasoby
Thread2
Wątek kożysta z zasobów

Mutex

```
#include <thread>

void threadFn(mutex &mtx){

lock_guard<mutex> lock(mtx);

//robi coś

}

int main(){

mutex mtx;

thread t1(threadFn,ref(mtx));

t1.join();

return 0;

}
```

Mutex

```
#include <thread>
void threadFn(mutex &mtx){
unique_lock<mutex> ul(mtx);
//robi coś
ul.unlock()
}
int main(){
mutex mtx;
thread t1(threadFn,ref(mtx));
t1.join();
return 0;
}
```