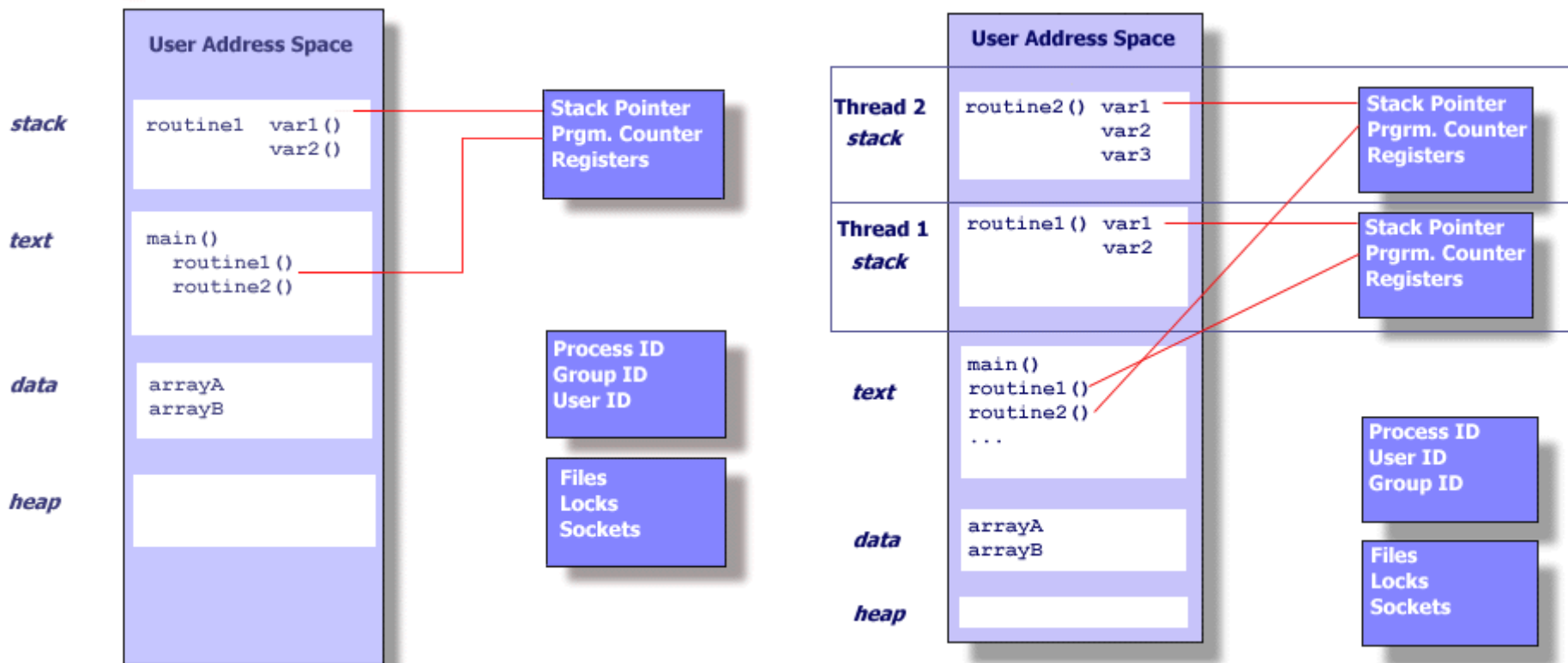




# Программирование потоков POSIX

Чопоров Сергей Викторович

# UNIX Process vs POSIX Threads



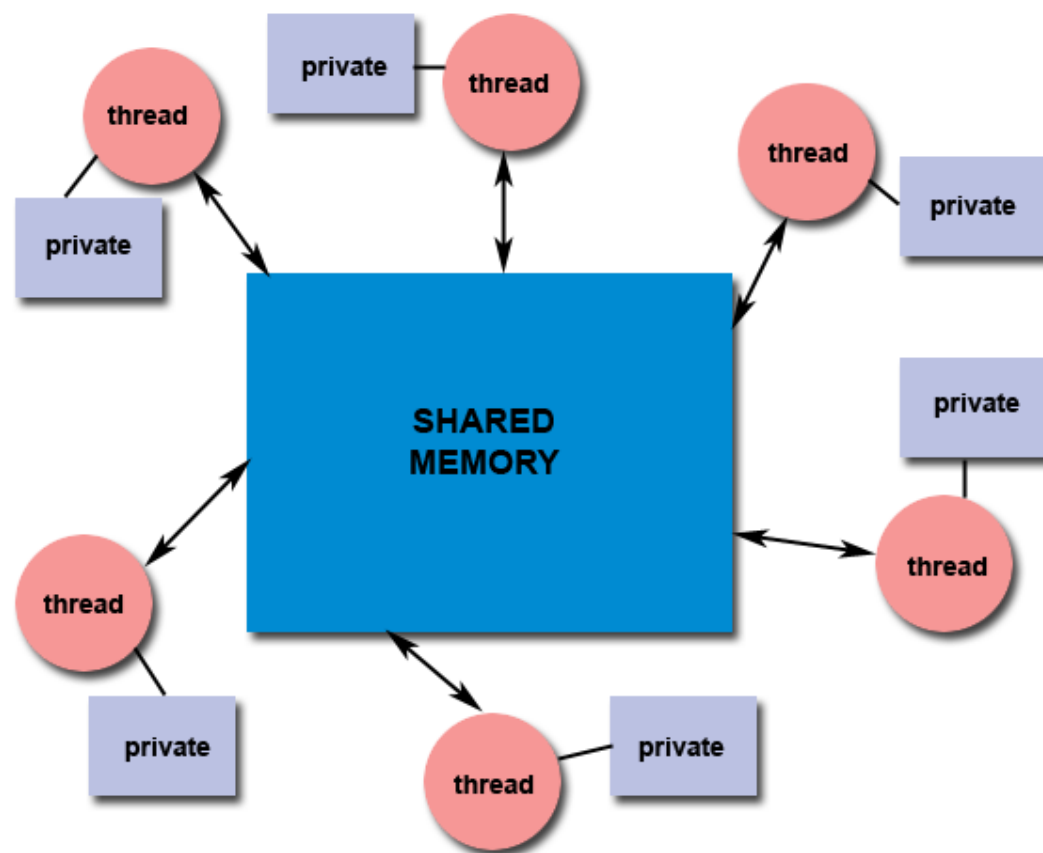
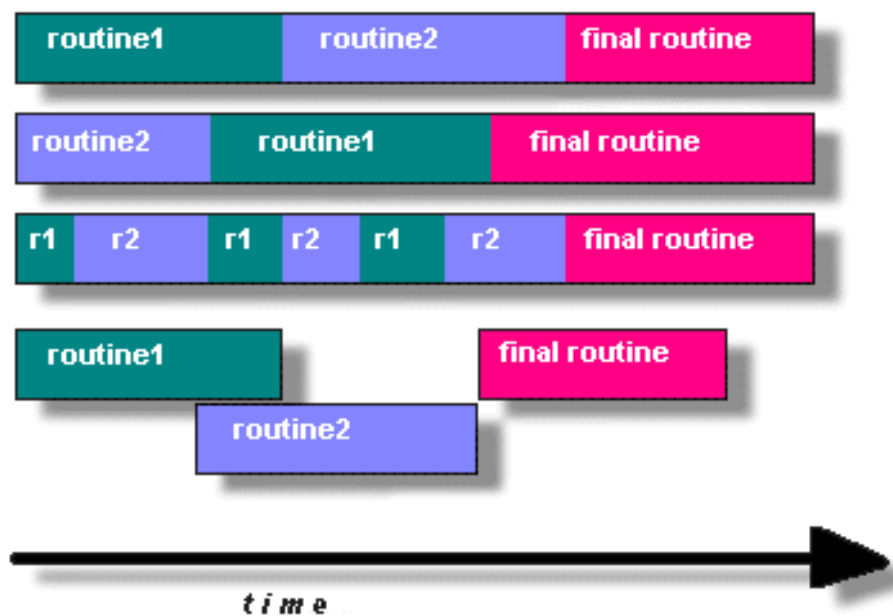
# Время создания 50 000 процессов / потоков

Платформа	Процесс			Поток POSIX		
	real	user	sys	real	user	sys
Intel 2.6 GHz Xeon E5-2670 (16 cores/node)	8,1	0,1	2,9	0,9	0,2	0,3
Intel 2.8 GHz Xeon 5660 (12 cores/node)	4,4	0,4	4,3	0,7	0,2	0,5
AMD 2.3 GHz Opteron (16 cores/node)	12,5	1,0	12,5	1,2	0,2	1,3
AMD 2.4 GHz Opteron (8 cores/node)	17,6	2,2	15,7	1,4	0,3	1,3
IBM 4.0 GHz POWER6 (8 cpus/node)	9,5	0,6	8,8	1,6	0,1	0,4

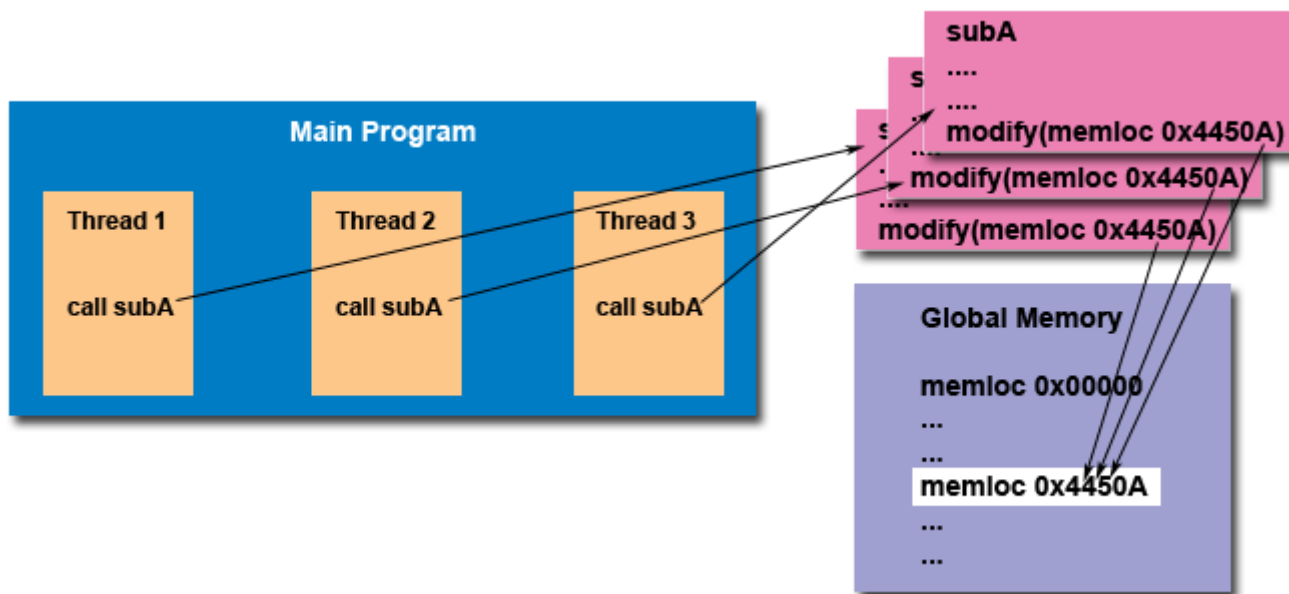
# Скорость обмена информацией

Платформа	MPI Shared Memory Bandwidth (GB/sec)	Pthreads Worst Case Memory-to-CPU Bandwidth (GB/sec)
Intel 2.6 GHz Xeon E5-2670	4,5	51,2
Intel 2.8 GHz Xeon 5660	5,6	32
AMD 2.3 GHz Opteron	1,8	5,3
AMD 2.4 GHz Opteron	1,2	5,3
IBM 1.9 GHz POWER5 p5-575	4,1	16

# Модель организации приложения



# Безопасность потоков POSIX



# Пример приложения

```
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>
#define NUM_THREADS 5
void *PrintHello(void *threadid)
{
    long tid;
    tid = (long)threadid;
    printf("Hello World! It's me, thread #%ld!\n", tid);
    pthread_exit(NULL);
}
int main(int argc, char *argv[])
{
    pthread_t threads[NUM_THREADS];
    int rc;
    long t;
    for(t=0;t<NUM_THREADS;t++){
        printf("In main: creating thread %ld\n", t);
        rc = pthread_create(&threads[t], NULL, PrintHello, (void *)t);
        if (rc){
            printf("ERROR; return code from pthread_create() is %d\n", rc);
            exit(-1);
        }
    }
    pthread_exit(NULL);
} // QMAKE_CXXFLAGS += -pthread LIBS += -pthread
```