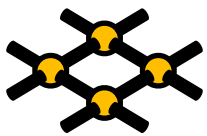
# Vláknové programování
# část II
# Kvíz

**Lukáš Hejmánek, Petr Holub**
**{xhejtman,hopet}@ics.muni.cz**



Laboratoř pokročilých síťových technologií

PV192
20010–03–17

# testandset

```
1
2
3  volatile int lock;
4
5  #define SET 1
6
7  int testandset(volatile int *l)
8  {
9          int old;
10         /* old = __sync_lock_test_and_set(l, SET);*/
11         asm("xchgl %1, %0" : "=r"(old) : "m" (*l), "0" (SET));
12         return old==1;
13 }
```

# lock/unlock

```
1  volatile int lock;
2  pthread_t t1, t2, t3;
3
4  void
5  my_lock()
6  {
7          while(testandset(&lock)) {
8                  sigsuspend(&set);
9                  /* lze pouzit
10                  * int i;
11                    sigwait(&set, &i) */
12         }
13 }
```

# lock/unlock

```
1  void
2  my_unlock()
3  {
4          pthread_t t;
5          lock=0;
6          t = pthread_self();
7          if(t != t1) {
8                  pthread_kill(t1, SIGCONT);
9          }
10         if(t != t2) {
11                 pthread_kill(t2, SIGCONT);
12         }
13         if(t != t3) {
14                 pthread_kill(t3, SIGCONT);
15         }
16         // nebo kill(getpid(), SIGCONT);
17 }
```

# main

```
1  #include <stdio.h>
2  #include <pthread.h>
3  #include <unistd.h>
4  #include <signal.h>
5
6  int
7  main(void)
8  {
9          sigemptyset(&set);
10         sigfillset(&set);
11         sigprocmask(SIG_BLOCK, &set, NULL);
12         signal(SIGCONT, sigcont);
13         sigdelset(&set, SIGCONT);
14         pthread_create(&t1, NULL, foo, NULL);
15         pthread_create(&t2, NULL, foo, NULL);
16         pthread_create(&t3, NULL, foo, NULL);
17         x=1;
18         pthread_join(t1, NULL);
19         pthread_join(t2, NULL);
20         pthread_join(t3, NULL);
21         return 0;
22 }
```