



Thread through threads

★ **Concepts**

★ Assembly instructions

★ Assignment #2

A concept of a **process**

*A **computer program** is a passive collection of instructions, while a **process** is the actual execution of those instructions.*

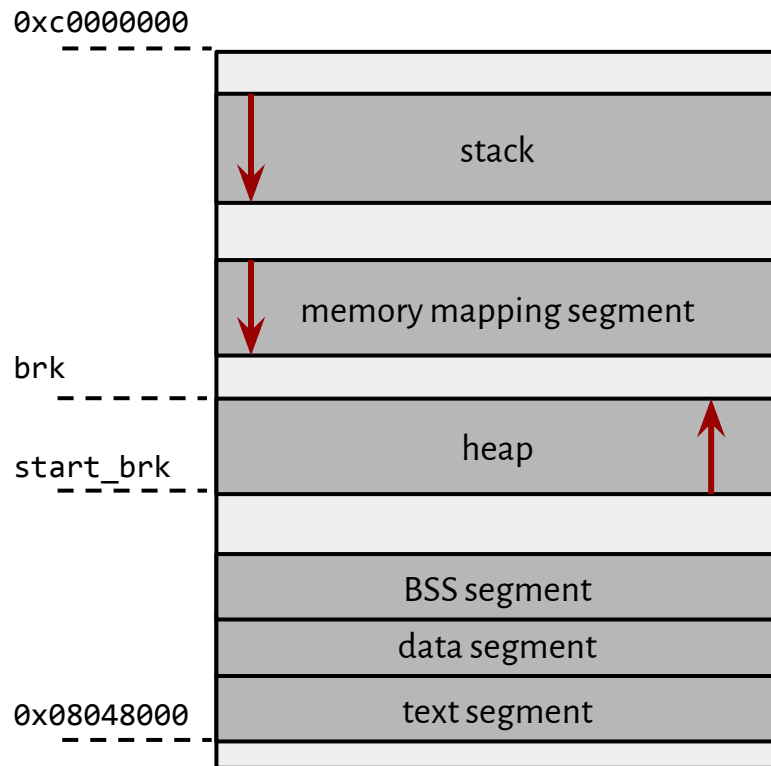
A concept of a **process**

a group of related resources

Processes are used to group resources together; [...]

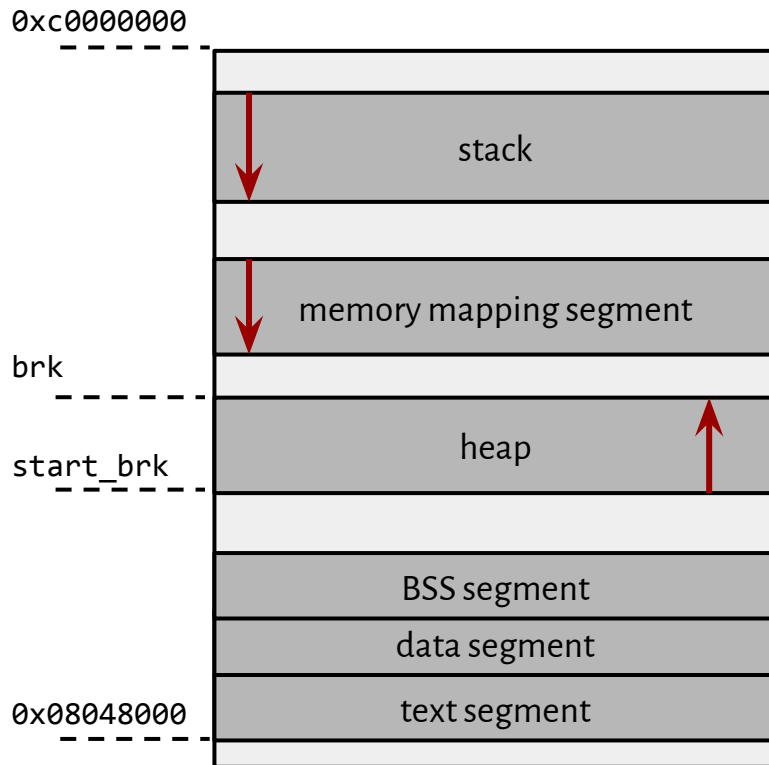
A concept of a **process**

a group of related resources



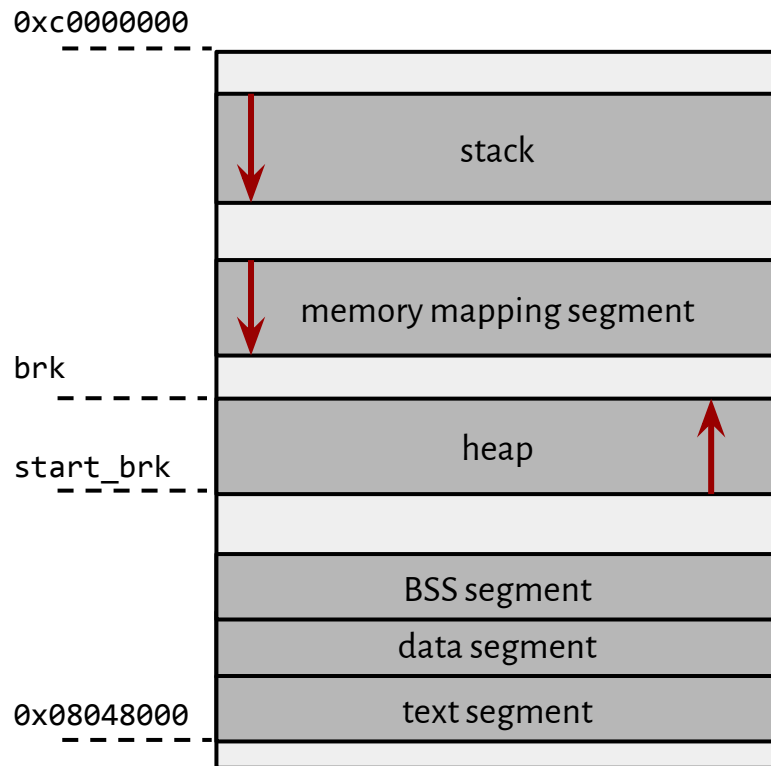
A concept of a **process**

a group of related resources



A concept of a **process**

a group of related resources



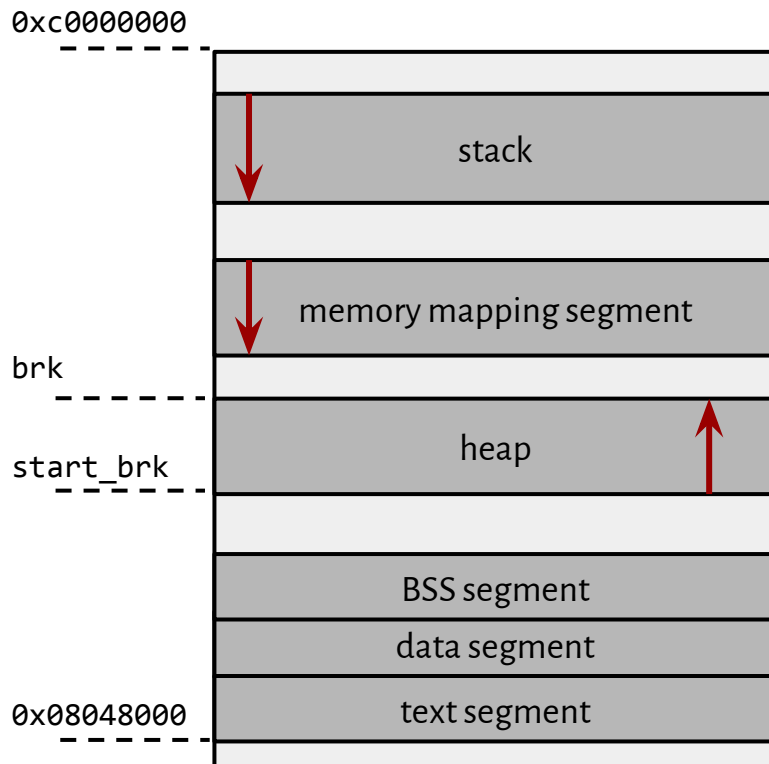
A concept of a **thread**

execution of instructions within a process

Processes are used to group resources together; threads are the entities scheduled for execution on the CPU.

A concept of a **thread**

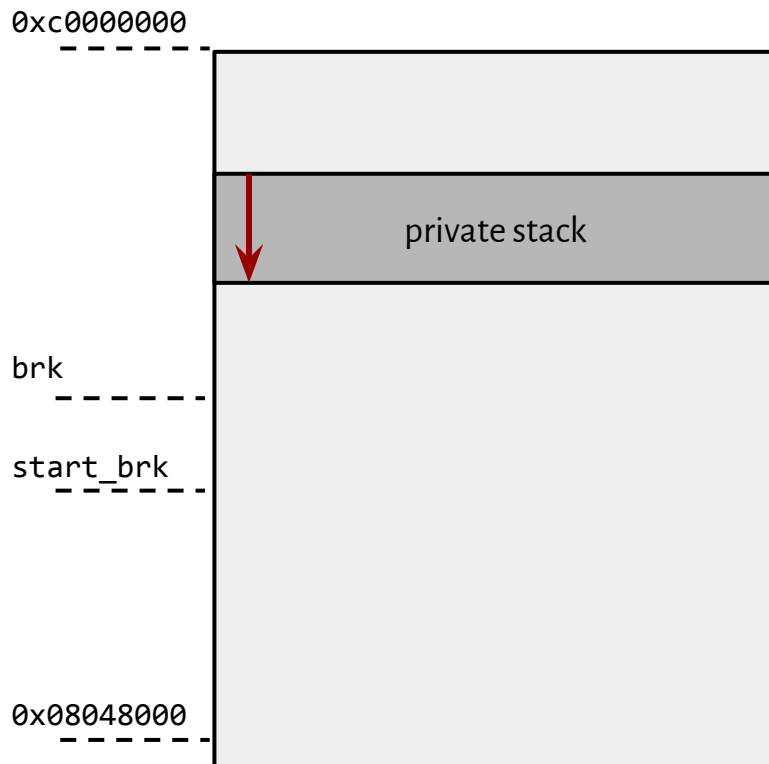
execution of instructions within a process



shared resources

A concept of a **thread**

execution of instructions within a process



private resources

Let's see!

[simple.c](#)

```
$ gcc -pthread simple.c -o simple; ./simple
```

```
$ cat /proc/29298/maps
```

Let's see!

[simple.c](#)

```
$ gcc -pthread simple.c -o simple; ./simple
```

```
$ cat /proc/29298/maps
```

```
00400000-00401000 r-xp 00000000 08:02 43010148 /home/inga/SO/3/simple
00600000-00601000 r--p 00000000 08:02 43010148 /home/inga/SO/3/simple
00601000-00602000 rw-p 00001000 08:02 43010148 /home/inga/SO/3/simple
019d0000-019f1000 rw-p 00000000 00:00 0 [heap]
7ff49b60f000-7ff49b610000 ---p 00000000 00:00 0
7ff49b610000-7ff49be10000 rw-p 00000000 00:00 0
[...]
7ff49c61d000-7ff49c61e000 rw-p 00000000 00:00 0
7ffffb6c9000-7ffffb6ea000 rw-p 00000000 00:00 0 [stack]
7ffffb779000-7ffffb77b000 r--p 00000000 00:00 0 [vvar]
7ffffb77b000-7ffffb77d000 r-xp 00000000 00:00 0 [vdso]
fffffffffff60000-fffffffffff601000 r-xp 00000000 00:00 0 [vsyscall]
```

My adventure with htop

```
1 [|||
2 [||
3 [|||
4 [|||||
Mem[|||||] 2.56G/3.75G
Swp[||] 68.4M/3.89G

1.3%] Tasks: 120, 380 thr; 1 running
0.7%] Load average: 1.36 0.63 0.39
1.3%] Uptime: 08:11:41
4.6%]

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
29337 inga 20 0 27920 4404 3128 R 2.0 0.1 0:23.89 htop
29298 inga 20 0 14716 756 672 T 0.0 0.0 0:03.51 ./simple
29299 inga 20 0 14716 756 672 T 0.0 0.0 0:01.75 ./simple
3319 inga 20 0 14872 1756 1600 S 0.0 0.0 0:00.00 gnome-pty-helper
3310 inga 20 0 770M 66444 36444 S 0.0 1.7 0:00.00 /usr/bin/python /usr/bin/terminator
3309 inga 20 0 770M 66444 36444 S 0.0 1.7 0:02.88 /usr/bin/python /usr/bin/terminator
3308 inga 20 0 770M 66444 36444 S 0.0 1.7 0:00.00 /usr/bin/python /usr/bin/terminator
2543 inga 20 0 62392 5500 5048 S 0.0 0.1 0:00.03 /usr/lib/x86_64-linux-gnu/gconf/gconfd-2
2488 inga 20 0 2692M 476M 129M S 3.3 12.4 21:01.13 /usr/lib/firefox/firefox
29963 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.00 /usr/lib/firefox/firefox
29925 inga 20 0 655M 63748 50672 S 0.0 1.6 0:00.42 /usr/lib/firefox/plugin-container /usr/lib/flashplugin-installer/libflashplayer.so
29930 inga 20 0 655M 63748 50672 S 0.0 1.6 0:00.00 /usr/lib/firefox/plugin-container /usr/lib/flashplugin-installer/libflashplayer.so
29929 inga 20 0 655M 63748 50672 S 0.0 1.6 0:00.00 /usr/lib/firefox/plugin-container /usr/lib/flashplugin-installer/libflashplayer.so
29928 inga 20 0 655M 63748 50672 S 0.0 1.6 0:00.00 /usr/lib/firefox/plugin-container /usr/lib/flashplugin-installer/libflashplayer.so
29927 inga 20 0 655M 63748 50672 S 0.0 1.6 0:00.00 /usr/lib/firefox/plugin-container /usr/lib/flashplugin-installer/libflashplayer.so
29918 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.00 /usr/lib/firefox/firefox
29915 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.00 /usr/lib/firefox/firefox
29466 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.02 /usr/lib/firefox/firefox
29062 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.05 /usr/lib/firefox/firefox
27775 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.00 /usr/lib/firefox/firefox
25668 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.00 /usr/lib/firefox/firefox
24943 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.00 /usr/lib/firefox/firefox
23104 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.26 /usr/lib/firefox/firefox
18465 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.00 /usr/lib/firefox/firefox
18338 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.00 /usr/lib/firefox/firefox
16848 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.00 /usr/lib/firefox/firefox
11881 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.00 /usr/lib/firefox/firefox
8890 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.00 /usr/lib/firefox/firefox
4281 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.00 /usr/lib/firefox/firefox
2868 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.00 /usr/lib/firefox/firefox
2651 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.00 /usr/lib/firefox/firefox
2650 inga 20 0 3069M 1176M 111M S 1.3 30.6 41:20.78 /usr/lib/firefox/plugin-container -greomni /usr/lib/firefox/omni.ja -appomni /usr/l
4321 inga 21 1 3069M 1176M 111M S 0.0 30.6 0:16.10 /usr/lib/firefox/plugin-container -greomni /usr/lib/firefox/omni.ja -appomni /us
2688 inga 20 0 3069M 1176M 111M S 0.0 30.6 0:00.00 /usr/lib/firefox/plugin-container -greomni /usr/lib/firefox/omni.ja -appomni /us
2685 inga 20 0 3069M 1176M 111M S 0.0 30.6 0:00.64 /usr/lib/firefox/plugin-container -greomni /usr/lib/firefox/omni.ja -appomni /us
2675 inga 20 0 3069M 1176M 111M S 0.0 30.6 0:00.00 /usr/lib/firefox/plugin-container -greomni /usr/lib/firefox/omni.ja -appomni /us
2674 inga 20 0 3069M 1176M 111M S 0.0 30.6 0:00.00 /usr/lib/firefox/plugin-container -greomni /usr/lib/firefox/omni.ja -appomni /us
2673 inga 20 0 3069M 1176M 111M S 0.0 30.6 0:00.00 /usr/lib/firefox/plugin-container -greomni /usr/lib/firefox/omni.ja -appomni /us

F1 Help F2 Setup F3 Search F4 Filter F5 Sorted F6 Collap F7 Nice F8 Nice F9 Kill F10 Quit
```

My adventure with htop

```
1 [|||
2 [||
3 [|||
4 [|||||
Mem[|||||||||||||||||||||||||||||||||] 2.56G/3.75G
Swp[|||||] 68.4M/3.89G

1.3%] Tasks: 120, 380 thr; 1 running
0.7%] Load average: 1.36 0.63 0.39
1.3%] Uptime: 08:11:41
4.6%]

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
29298 inga 20 0 14716 756 672 T 0.0 0.0 0:03.51 htop
29299 inga 20 0 14716 756 672 T 0.0 0.0 0:01.75 ./simple
3319 inga 20 0 14872 1756 1600 S 0.0 0.0 0:00.00 ./simple
3310 inga 20 0 770M 66444 36444 S 0.0 1.7 0:00.00 gnome-pty-helper
3309 inga 20 0 770M 66444 36444 S 0.0 1.7 0:02.88 /usr/bin/python /usr/bin/terminator
3308 inga 20 0 770M 66444 36444 S 0.0 1.7 0:00.00 /usr/bin/python /usr/bin/terminator
2543 inga 20 0 62392 5500 5048 S 0.0 0.1 0:00.03 /usr/lib/x86_64-linux-gnu/gconf/gconfd-2
2488 inga 20 0 2692M 476M 129M S 3.3 12.4 21:01.13 /usr/lib/firefox/firefox
29963 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.00 /usr/lib/firefox/firefox
29925 inga 20 0 655M 63748 50672 S 0.0 1.6 0:00.42 /usr/lib/firefox/plugin-container /usr/lib/flashplugin-installer/libflashplayer.so
29930 inga 20 0 655M 63748 50672 S 0.0 1.6 0:00.00 /usr/lib/firefox/plugin-container /usr/lib/flashplugin-installer/libflashplayer.so
29929 inga 20 0 655M 63748 50672 S 0.0 1.6 0:00.00 /usr/lib/firefox/plugin-container /usr/lib/flashplugin-installer/libflashplayer.so
29928 inga 20 0 655M 63748 50672 S 0.0 1.6 0:00.00 /usr/lib/firefox/plugin-container /usr/lib/flashplugin-installer/libflashplayer.so
29927 inga 20 0 655M 63748 50672 S 0.0 1.6 0:00.00 /usr/lib/firefox/plugin-container /usr/lib/flashplugin-installer/libflashplayer.so
29918 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.00 /usr/lib/firefox/firefox
29915 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.00 /usr/lib/firefox/firefox
29466 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.02 /usr/lib/firefox/firefox
29062 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.05 /usr/lib/firefox/firefox
27775 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.00 /usr/lib/firefox/firefox
25668 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.00 /usr/lib/firefox/firefox
24943 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.00 /usr/lib/firefox/firefox
23104 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.26 /usr/lib/firefox/firefox
18465 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.00 /usr/lib/firefox/firefox
18338 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.00 /usr/lib/firefox/firefox
16848 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.00 /usr/lib/firefox/firefox
11881 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.00 /usr/lib/firefox/firefox
8890 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.00 /usr/lib/firefox/firefox
4281 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.00 /usr/lib/firefox/firefox
2868 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.00 /usr/lib/firefox/firefox
2651 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.00 /usr/lib/firefox/firefox
2650 inga 20 0 3069M 1176M 111M S 1.3 30.6 41:20.78 /usr/lib/firefox/plugin-container -greomni /usr/lib/firefox/omni.ja -appomni /usr/l
4321 inga 21 1 3069M 1176M 111M S 0.0 30.6 0:16.10 /usr/lib/firefox/plugin-container -greomni /usr/lib/firefox/omni.ja -appomni /us
2688 inga 20 0 3069M 1176M 111M S 0.0 30.6 0:00.00 /usr/lib/firefox/plugin-container -greomni /usr/lib/firefox/omni.ja -appomni /us
2685 inga 20 0 3069M 1176M 111M S 0.0 30.6 0:00.64 /usr/lib/firefox/plugin-container -greomni /usr/lib/firefox/omni.ja -appomni /us
2675 inga 20 0 3069M 1176M 111M S 0.0 30.6 0:00.00 /usr/lib/firefox/plugin-container -greomni /usr/lib/firefox/omni.ja -appomni /us
2674 inga 20 0 3069M 1176M 111M S 0.0 30.6 0:00.00 /usr/lib/firefox/plugin-container -greomni /usr/lib/firefox/omni.ja -appomni /us
2673 inga 20 0 3069M 1176M 111M S 0.0 30.6 0:00.00 /usr/lib/firefox/plugin-container -greomni /usr/lib/firefox/omni.ja -appomni /us

F1 Help F2 Setup F3 Search F4 Filter F5 Sorted F6 Collap F7 Nice F8 Nice + F9 Kill F10 Quit
```

My adventure with htop

```
1  [||] 1.3%] Tasks: 120, 380 thr; 1 running
2  [||] 0.7%] Load average: 1.36 0.63 0.39
3  [||] 1.3%] Uptime: 08:11:41
4  [||||] 4.6%]
Mem[|||||] 2.56G/3.75G]
Swp[||] 68.4M/3.89G]

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
20337 inga 20 0 27920 4404 3128 R 2.0 0.1 0:23.89 htop
29298 inga 20 0 14716 756 672 T 0.0 0.0 0:03.51 ./simple
29299 inga 20 0 14716 756 672 T 0.0 0.0 0:01.75 ./simple
3319 inga 20 0 14872 1756 1600 S 0.0 0.0 0:00.00 gnome-pty-helper
3310 inga 20 0 770M 66444 36444 S 0.0 1.7 0:00.00 /usr/bin/python /usr/bin/terminator
3309 inga 20 0 770M 66444 36444 S 0.0 1.7 0:02.88 /usr/bin/python /usr/bin/terminator
3308 inga 20 0 770M 66444 36444 S 0.0 1.7 0:00.00 /usr/bin/python /usr/bin/terminator
2543 inga 20 0 62392 5500 5048 S 0.0 0.1 0:00.03 /usr/lib/x86_64-linux-gnu/gconf/gconfd-2
2488 inga 20 0 2692M 476M 129M S 3.3 12.4 1:01.13 /usr/lib/firefox/firefox
```



```
29298 inga 20 0 14716 756 672 T 0.0 0.0 0:03.51 ./simple
29299 inga 20 0 14716 756 672 T 0.0 0.0 0:01.75 ./simple
```

Why PIDs are different?

What are threads in Linux?

processes that share certain resources

What are threads in Linux?

***tasks** that share certain resources*

What are threads in Linux?

tasks that share certain resources

```
$ strace ./simple
```

```
clone(child_stack=0x7f18a5423ff0,  
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETT  
LS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, parent_tidptr=0x7f18a54249d0,  
tls=0x7f18a5424700, child_tidptr=0x7f18a54249d0) = 29299  
child thread pid is 29299
```

What are threads in Linux?

tasks that share certain resources

```
$ strace ./simple
```

```
clone(child_stack=0x7f18a5423ff0,  
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETT  
LS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, parent_tidptr=0x7f18a54249d0,  
tls=0x7f18a5424700, child_tidptr=0x7f18a54249d0) = 29299  
child thread pid is 29299
```

```
$ cat /proc/29298/maps
```

```
$ cat /proc/29299/maps
```

What are threads in Linux?

tasks that share certain resources

```
$ strace ./simple
```

```
clone(child_stack=0x7f18a5423ff0,  
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETT  
LS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARPID, parent_tidptr=0x7f18a54249d0,  
tls=0x7f18a5424700, child_tidptr=0x7f18a54249d0) = 29299  
child thread pid is 29299
```

```
$ cat /proc/29298/maps
```

```
$ cat /proc/29299/maps
```

```
$ ls -ls /proc/29299
```

My adventure with htop

```
 1 [||| 1.3%] Tasks: 120, 380 thr; 1 running
 2 [|| 0.7%] Load average: 1.36 0.63 0.39
 3 [||| 1.3%] Uptime: 08:11:41
 4 [|||| 4.6%]
Mem[|||||] 2.56G/3.75G
Swp[|||] 68.4M/3.89G

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
29337 inga 20 0 27920 4404 3128 R 2.0 0.1 0:23.89 htop
29298 inga 20 0 14716 756 672 T 0.0 0.0 0:03.51 ./simple
29299 inga 20 0 14716 756 672 T 0.0 0.0 0:01.75 ./simple
3319 inga 20 0 14872 1756 1600 S 0.0 0.0 0:00.00 gnome-pty-helper
3310 inga 20 0 770M 66444 36444 S 0.0 1.7 0:00.00 /usr/bin/python /usr/bin/terminator
3309 inga 20 0 770M 66444 36444 S 0.0 1.7 0:02.88 /usr/bin/python /usr/bin/terminator
3308 inga 20 0 770M 66444 36444 S 0.0 1.7 0:00.00 /usr/bin/python /usr/bin/terminator
2543 inga 20 0 62392 5500 5048 S 0.0 0.1 0:00.03 /usr/lib/x86_64-linux-gnu/gconf/gconfd-2
2488 inga 20 0 2692M 476M 129M S 3.3 12.4 21:01.13 /usr/lib/firefox/firefox
29963 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.00 /usr/lib/firefox/firefox
29925 inga 20 0 655M 63748 50672 S 0.0 1.6 0:00.42 /usr/lib/firefox/plugin-container /usr/lib/flashplugin-installer/libflashplayer.so
29930 inga 20 0 655M 63748 50672 S 0.0 1.6 0:00.00 /usr/lib/firefox/plugin-container /usr/lib/flashplugin-installer/libflashplayer.so
29929 inga 20 0 655M 63748 50672 S 0.0 1.6 0:00.00 /usr/lib/firefox/plugin-container /usr/lib/flashplugin-installer/libflashplayer.so
29928 inga 20 0 655M 63748 50672 S 0.0 1.6 0:00.00 /usr/lib/firefox/plugin-container /usr/lib/flashplugin-installer/libflashplayer.so
29927 inga 20 0 655M 63748 50672 S 0.0 1.6 0:00.00 /usr/lib/firefox/plugin-container /usr/lib/flashplugin-installer/libflashplayer.so
29918 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.00 /usr/lib/firefox/firefox
29915 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.00 /usr/lib/firefox/firefox
29466 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.02 /usr/lib/firefox/firefox
29062 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.05 /usr/lib/firefox/firefox
27775 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.00 /usr/lib/firefox/firefox
25668 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.00 /usr/lib/firefox/firefox
24943 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.00 /usr/lib/firefox/firefox
23104 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.26 /usr/lib/firefox/firefox
18465 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.00 /usr/lib/firefox/firefox
18338 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.00 /usr/lib/firefox/firefox
16848 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.00 /usr/lib/firefox/firefox
11881 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.00 /usr/lib/firefox/firefox
8890 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.00 /usr/lib/firefox/firefox
4281 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.00 /usr/lib/firefox/firefox
2868 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.00 /usr/lib/firefox/firefox
2651 inga 20 0 2692M 476M 129M S 0.0 12.4 0:00.00 /usr/lib/firefox/firefox
2650 inga 20 0 3069M 1176M 111M S 1.3 30.6 41:20.78 /usr/lib/firefox/plugin-container -greomni /usr/lib/firefox/omni.ja -appomni /usr/l
4321 inga 21 1 3069M 1176M 111M S 0.0 30.6 0:16.10 /usr/lib/firefox/plugin-container -greomni /usr/lib/firefox/omni.ja -appomni /us
2688 inga 20 0 3069M 1176M 111M S 0.0 30.6 0:00.00 /usr/lib/firefox/plugin-container -greomni /usr/lib/firefox/omni.ja -appomni /us
2685 inga 20 0 3069M 1176M 111M S 0.0 30.6 0:00.64 /usr/lib/firefox/plugin-container -greomni /usr/lib/firefox/omni.ja -appomni /us
2675 inga 20 0 3069M 1176M 111M S 0.0 30.6 0:00.00 /usr/lib/firefox/plugin-container -greomni /usr/lib/firefox/omni.ja -appomni /us
2674 inga 20 0 3069M 1176M 111M S 0.0 30.6 0:00.00 /usr/lib/firefox/plugin-container -greomni /usr/lib/firefox/omni.ja -appomni /us
2673 inga 20 0 3069M 1176M 111M S 0.0 30.6 0:00.00 /usr/lib/firefox/plugin-container -greomni /usr/lib/firefox/omni.ja -appomni /us

F1 Help F2 Setup F3 Search F4 Filter F5 Sorted F6 Collap F7 Nice F8 Nice F9 Kill F10 Quit
```



★ Concepts

★ **Assembly instructions**

★ Assignment #2

Assembly - atomic instructions

**lock **

<instr>

Assembly - atomic instructions

**lock **

<instr>

xchg **dst**, **src**

dst' = **src**

src' = **dst**

Assembly - atomic instructions

**lock **
<instr>

xchg **dst**, **src**

dst' = **src**
src' = **dst**

cmpxchg **dst**, **src**

if **rax** == **dst**:
 dst' = **src**
else:
 rax' = **dst**

Scenario

```
void *inc_thread(void *);
```

```
typedef struct {  
    int *value;  
    int count;  
} thread_data_t;
```

```
$ cd /home/students/inf/PUBLIC/S0/scenariusze/3
```

```
$ make
```

```
$ ./inc_thread_test_naive 3 1000000
```

Scenario

```
global inc_thread
```

lock

```
section .text
```

```
align 8
```

```
inc_thread:
```

```
    mov     rsi, [rdi]      ; value  
    mov     ecx, [rdi + 8] ; count  
    jmp     end
```

```
count_loop:
```

```
    inc     dword [rsi]    ; ++*value
```

```
count_test:
```

```
    sub     ecx, 1         ; --count  
    jge    loop           ; skok, gdy count >= 0  
    xor     eax, eax      ; return NULL  
    ret
```

Scenario

```
global inc_thread
```

```
section .text
```

```
align 8
```

```
inc_thread:
```

```
    mov     rsi, [rdi]      ; value  
    mov     ecx, [rdi + 8] ; count  
    jmp     end
```

```
count_loop:
```

```
    inc     dword [rsi]    ; ++*value
```

```
count_test:
```

```
    sub     ecx, 1         ; --count  
    jge    loop           ; skok, gdy count >= 0  
    xor     eax, eax      ; return NULL  
    ret
```

lock

xchg

Spinlock

wait:

```
    if (spinlock == 0) {  
        spinlock = 1  
        go to critical_section  
        spinlock = 0  
    } else {  
        go to wait  
    }
```

critical_section:

...

Spinlock

```
wait:  
    if (spinlock == 0) {  
        spinlock = 1  
        go to critical_section  
        spinlock = 0  
    } else {  
        go to wait  
    }  
  
critical_section:  
...
```

Spinlock

```
wait:  
    if (spinlock == 0) {  
        spinlock = 1  
        go to critical_section  
        spinlock = 0  
    } else {  
        go to wait  
    }  
  
critical_section:  
...
```

Spinlock

```
wait:  
    if (spinlock == 0) {  
        spinlock = 1  
        go to critical_section  
        spinlock = 0  
    } else {  
        go to wait  
    }
```

```
critical_section:  
...
```

Spinlock

wait:

```
if (spinlock == 0) {  
    spinlock = 1  
    go to critical_section  
    spinlock = 0  
} else {  
    go to wait  
}
```

critical_section:

...

What does need to be atomic?

Spinlock

wait:

```
    if (spinlock == 0) {  
        spinlock = 1  
        go to critical_section  
        spinlock = 0  
    } else {  
        go to wait  
    }
```

} atomic

critical_section:

...

Scenario

```
global inc_thread
```

```
section .text
```

```
align 8
```

```
inc_thread:
```

```
    mov     rsi, [rdi]      ; value
```

```
    mov     ecx, [rdi + 8] ; count
```

```
    jmp     end
```

```
count_loop:
```

```
    inc     dword [rsi]    ; ++*value
```

```
count_test:
```

```
    sub     ecx, 1         ; --count
```

```
    jge    loop           ; skok, gdy count >= 0
```

```
    xor     eax, eax      ; return NULL
```

```
    ret
```

lock

xchg

cmpxchg

Scenario

```
global inc_thread
```

```
section .text
```

```
align 8
```

```
inc_thread:
```

```
    mov     rsi, [rdi]      ; value  
    mov     ecx, [rdi + 8]  ; count  
    jmp     end
```

```
count_loop:
```

```
    inc     dword [rsi]    ; ++*value
```

```
count_test:
```

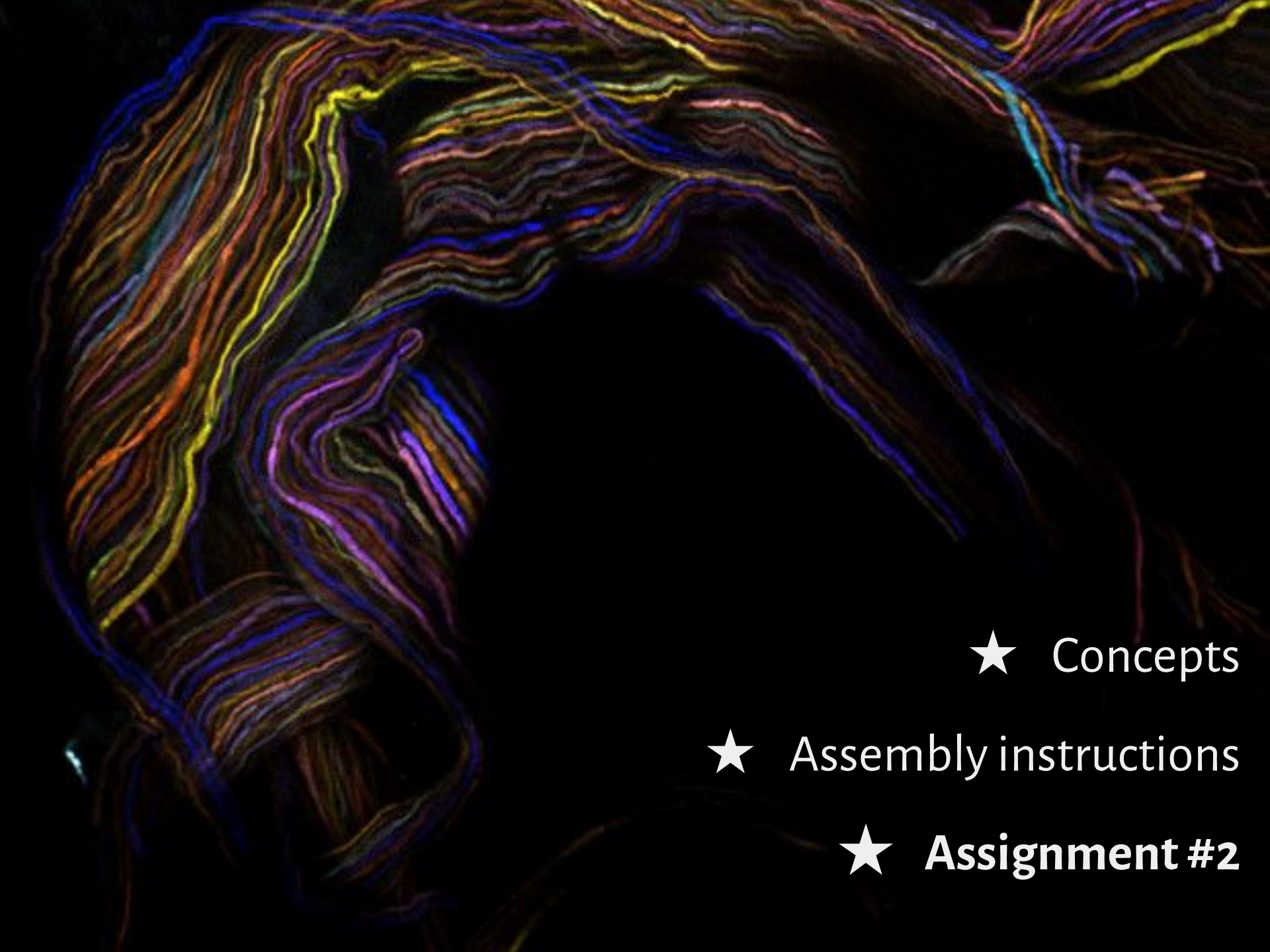
```
    sub     ecx, 1         ; --count  
    jge    loop           ; skok, gdy count >= 0  
    xor     eax, eax      ; return NULL  
    ret
```

lock

xchg

cmpxchg

bts, btr



★ Concepts

★ Assembly instructions

★ **Assignment #2**

Assignment #2

Zaimplementuj w asemblerze x86_64 moduł symulujący działanie **sieci euronowej**.

Sieć składa się z N euronów, które są numerowane od 0 do $(N - 1)$.

Eurony działają równolegle – każdy euron jest uruchamiany w osobnym wątku.

Moduł będzie używany z języka C i ma udostępniać funkcję widzianą jako:

```
uint64_t euron(uint64_t n, char const *prog);
```

- Parametr **n** zawiera numer euronu.
- Parametr **prog** jest wskaźnikiem na napis ASCII i opisuje obliczenie, jakie ma wykonać euron.

Przykład poprawnego obliczenia dla $N = 2$ (jako napis w języku C):

```
"01234n+P56789E-+D+*G*1n-+S2ED+E1-+75+-BC"
```

Assignment #2

Przykład poprawnego obliczenia dla $N = 2$ (jako napis w języku C):

```
"01234n+P56789E-+D+*G*1n-+S2ED+E1-+75+-BC"
```

'+'	zdejmij dwie wartości ze stosu, oblicz ich sumę i wstaw wynik na stos;
'*'	zdejmij dwie wartości ze stosu, oblicz ich iloczyn i wstaw wynik na stos;
'-'	zaneguj arytmetycznie wartość na wierzchołku stosu;
'0-9'	wstaw na stos odpowiednio liczbę 0 do 9;
'n'	wstaw na stos numer euronu;

Assignment #2

Przykład poprawnego obliczenia dla $N = 2$ (jako napis w języku C):


"01234n+P56789E-+D+*G*1n-+S2ED+E1-+75+-BC"

B	BRANCH: zdejmij wartość ze stosu, v i spójrz na wierzchołek stosu, wartość w : jeśli $w < 0$, potraktuj v jako liczbę w kodzie uzupełnieniowym do dwójki i przesunij się o tyle operacji;
C	CLEAN: zdejmij wartość ze stosu;
D	DUPLICATE: zduplikuj wartość na wierzchu stosu;
E	EXCHANGE: zamień miejscami dwie wartości na wierzchu stosu;
G	GET: wstaw na stos wartość uzyskaną z wywołania funkcji <code>uint64_t get_value(uint64_t n);</code>
P	PUT: zdejmij wartość ze stosu (w) i wywołaj funkcję: <code>void put_value(uint64_t n, uint64_t w);</code>
S	SYNCHRONIZE: zdejmij wartość ze stosu, potraktuj ją jako numer euronu m , czekaj na operację 'S' euronu m ze zdjętym ze stosu numerem euronu n i zamień wartości na wierzchołkach stosów euronów m i n .

Assignment #2

Przykład poprawnego obliczenia dla $N = 2$ (jako napis w języku C):

"01234n+P56789E-+D+*G*1n-+S2ED+E1-+75+-BC"




```
uint64_t get_value(uint64_t n) {  
    assert(n < N);  
    return n + 1;  
}
```

```
void put_value(uint64_t n, uint64_t v) {  
    assert(n < N);  
    assert(v == n + 4);  
}
```


Assignment #2

Przykład poprawnego obliczenia dla $N = 2$ (jako napis w języku C):

"01234n+P56789E-+D+*G*1n-+S2ED+E1-+75+-BC"



0
1
2
3
4

```
uint64_t get_value(uint64_t n) {  
    assert(n < N);  
    return n + 1;  
}
```

```
void put_value(uint64_t n, uint64_t v) {  
    assert(n < N);  
    assert(v == n + 4);  
}
```

Assignment #2

Przykład poprawnego obliczenia dla $N = 2$ (jako napis w języku C):

"01234n+P56789E-+D+*G*1n-+S2ED+E1-+75+-BC"



0
1
2
3
4
0/1

```
uint64_t get_value(uint64_t n) {  
    assert(n < N);  
    return n + 1;  
}
```

```
void put_value(uint64_t n, uint64_t v) {  
    assert(n < N);  
    assert(v == n + 4);  
}
```

Assignment #2

Przykład poprawnego obliczenia dla $N = 2$ (jako napis w języku C):

"01234n+P56789E-+D+*G*1n-+S2ED+E1-+75+-BC"



0
1
2
3
4/5

```
uint64_t get_value(uint64_t n) {  
    assert(n < N);  
    return n + 1;  
}
```

```
void put_value(uint64_t n, uint64_t v) {  
    assert(n < N);  
    assert(v == n + 4);  
}
```

Assignment #2

Przykład poprawnego obliczenia dla $N = 2$ (jako napis w języku C):

"01234n+P56789E-+D+*G*1n-+S2ED+E1-+75+-BC"



0
1
2
3

```
uint64_t get_value(uint64_t n) {  
    assert(n < N);  
    return n + 1;  
}
```

```
void put_value(uint64_t n, uint64_t v) {  
    assert(n < N);  
    assert(v == n + 4);  
}
```

Assignment #2

Przykład poprawnego obliczenia dla $N = 2$ (jako napis w języku C):

"01234n+P56789E-+D+*G*1n-+S2ED+E1-+75+-BC"



0
1
2
3
5
6
7
8
9

```
uint64_t get_value(uint64_t n) {  
    assert(n < N);  
    return n + 1;  
}
```

```
void put_value(uint64_t n, uint64_t v) {  
    assert(n < N);  
    assert(v == n + 4);  
}
```

Assignment #2

Przykład poprawnego obliczenia dla $N = 2$ (jako napis w języku C):

"01234n+P56789E-+D+*G*1n-+S2ED+E1-+75+-BC"



0
1
2
3
5
6
7
9
8

```
uint64_t get_value(uint64_t n) {  
    assert(n < N);  
    return n + 1;  
}
```

```
void put_value(uint64_t n, uint64_t v) {  
    assert(n < N);  
    assert(v == n + 4);  
}
```

Assignment #2

Przykład poprawnego obliczenia dla $N = 2$ (jako napis w języku C):

"01234n+P56789E-+D+*G*1n-+S2ED+E1-+75+-BC"
↑

0
1
2
3
5
6
7
9
-8


```
uint64_t get_value(uint64_t n) {  
    assert(n < N);  
    return n + 1;  
}
```

```
void put_value(uint64_t n, uint64_t v) {  
    assert(n < N);  
    assert(v == n + 4);  
}
```

Assignment #2

Przykład poprawnego obliczenia dla $N = 2$ (jako napis w języku C):

"01234n+P56789E-+**D**+*G*1n-+S2ED+E1-+75+-BC"



0
1
2
3
5
6
7
1

```
uint64_t get_value(uint64_t n) {  
    assert(n < N);  
    return n + 1;  
}
```

```
void put_value(uint64_t n, uint64_t v) {  
    assert(n < N);  
    assert(v == n + 4);  
}
```


Assignment #2

Przykład poprawnego obliczenia dla $N = 2$ (jako napis w języku C):

"01234n+P56789E-+D+*G*1n-+S2ED+E1-+75+-BC"
↑

0
1
2
3
5
6
7
1
1

```
uint64_t get_value(uint64_t n) {  
    assert(n < N);  
    return n + 1;  
}
```

```
void put_value(uint64_t n, uint64_t v) {  
    assert(n < N);  
    assert(v == n + 4);  
}
```

Assignment #2

Przykład poprawnego obliczenia dla $N = 2$ (jako napis w języku C):

"01234n+P56789E-+D+*G*1n-+S2ED+E1-+75+-BC"



0
1
2
3
5
6
7
2

```
uint64_t get_value(uint64_t n) {  
    assert(n < N);  
    return n + 1;  
}
```

```
void put_value(uint64_t n, uint64_t v) {  
    assert(n < N);  
    assert(v == n + 4);  
}
```

Assignment #2

Przykład poprawnego obliczenia dla $N = 2$ (jako napis w języku C):

"01234n+P56789E-+D+*G*1n-+S2ED+E1-+75+-BC"
↑

0
1
2
3
5
6
14

```
uint64_t get_value(uint64_t n) {  
    assert(n < N);  
    return n + 1;  
}
```

```
void put_value(uint64_t n, uint64_t v) {  
    assert(n < N);  
    assert(v == n + 4);  
}
```

Assignment #2

Przykład poprawnego obliczenia dla $N = 2$ (jako napis w języku C):

"01234n+P56789E-+D+*G*1n-+S2ED+E1-+75+-BC"



0
1
2
3
5
6
14
1/2


```
uint64_t get_value(uint64_t n) {  
    assert(n < N);  
    return n + 1;  
}
```

```
void put_value(uint64_t n, uint64_t v) {  
    assert(n < N);  
    assert(v == n + 4);  
}
```

Assignment #2

Przykład poprawnego obliczenia dla $N = 2$ (jako napis w języku C):

"01234n+P56789E-+D+*G***1**n-+S2ED+E1-+75+-BC"



0
1
2
3
5
6
14/28


```
uint64_t get_value(uint64_t n) {  
    assert(n < N);  
    return n + 1;  
}
```

```
void put_value(uint64_t n, uint64_t v) {  
    assert(n < N);  
    assert(v == n + 4);  
}
```

Assignment #2

Przykład poprawnego obliczenia dla $N = 2$ (jako napis w języku C):

"01234n+P56789E-+D+*G*1n-+S2ED+E1-+75+-BC"



0
1
2
3
5
6
14/28
1

```
uint64_t get_value(uint64_t n) {  
    assert(n < N);  
    return n + 1;  
}
```

```
void put_value(uint64_t n, uint64_t v) {  
    assert(n < N);  
    assert(v == n + 4);  
}
```

Assignment #2

Przykład poprawnego obliczenia dla $N = 2$ (jako napis w języku C):

"01234n+P56789E-+D+*G*1n-+S2ED+E1-+75+-BC"



0
1
2
3
5
6
14/28
1
0/1

```
uint64_t get_value(uint64_t n) {  
    assert(n < N);  
    return n + 1;  
}
```

```
void put_value(uint64_t n, uint64_t v) {  
    assert(n < N);  
    assert(v == n + 4);  
}
```

Assignment #2

Przykład poprawnego obliczenia dla $N = 2$ (jako napis w języku C):

"01234n+P56789E-+D+*G*1n-+S2ED+E1-+75+-BC"
↑

0
1
2
3
5
6
14/28
1
0/-1


```
uint64_t get_value(uint64_t n) {  
    assert(n < N);  
    return n + 1;  
}
```

```
void put_value(uint64_t n, uint64_t v) {  
    assert(n < N);  
    assert(v == n + 4);  
}
```


Assignment #2

Przykład poprawnego obliczenia dla $N = 2$ (jako napis w języku C):

"01234n+P56789E-+D+*G*1n-+S2ED+E1-+75+-BC"



0
1
2
3
5
6
14/28
1/0

```
uint64_t get_value(uint64_t n) {  
    assert(n < N);  
    return n + 1;  
}
```

```
void put_value(uint64_t n, uint64_t v) {  
    assert(n < N);  
    assert(v == n + 4);  
}
```

Assignment #2

Przykład poprawnego obliczenia dla $N = 2$ (jako napis w języku C):

"01234n+P56789E-+D+*G*1n-+S2ED+E1-+75+-BC"



0
1
2
3
5
6
28/14

```
uint64_t get_value(uint64_t n) {  
    assert(n < N);  
    return n + 1;  
}
```

```
void put_value(uint64_t n, uint64_t v) {  
    assert(n < N);  
    assert(v == n + 4);  
}
```

Assignment #2

Przykład poprawnego obliczenia dla $N = 2$ (jako napis w języku C):

"01234n+P56789E-+D+*G*1n-+S2ED+E1-+75+-BC"



0
1
2
3
5
6
28/14
2

```
uint64_t get_value(uint64_t n) {  
    assert(n < N);  
    return n + 1;  
}
```

```
void put_value(uint64_t n, uint64_t v) {  
    assert(n < N);  
    assert(v == n + 4);  
}
```

Assignment #2

Przykład poprawnego obliczenia dla $N = 2$ (jako napis w języku C):

"01234n+P56789E-+D+*G*1n-+S2ED+E1-+75+-BC"



0
1
2
3
5
6
2
28/14

```
uint64_t get_value(uint64_t n) {  
    assert(n < N);  
    return n + 1;  
}
```

```
void put_value(uint64_t n, uint64_t v) {  
    assert(n < N);  
    assert(v == n + 4);  
}
```

Assignment #2

Przykład poprawnego obliczenia dla $N = 2$ (jako napis w języku C):

"01234n+P56789E-+D+*G*1n-+S2ED+E1-+75+-BC"



0
1
2
3
5
6
2
28/14
28/14

```
uint64_t get_value(uint64_t n) {  
    assert(n < N);  
    return n + 1;  
}
```

```
void put_value(uint64_t n, uint64_t v) {  
    assert(n < N);  
    assert(v == n + 4);  
}
```

Assignment #2

Przykład poprawnego obliczenia dla $N = 2$ (jako napis w języku C):

"01234n+P56789E-+D+*G*1n-+S2ED+E1-+75+-BC"




0
1
2
3
5
6
2
56/28

```
uint64_t get_value(uint64_t n) {  
    assert(n < N);  
    return n + 1;  
}
```

```
void put_value(uint64_t n, uint64_t v) {  
    assert(n < N);  
    assert(v == n + 4);  
}
```

Assignment #2

Przykład poprawnego obliczenia dla $N = 2$ (jako napis w języku C):

"01234n+P56789E-+D+*G*1n-+S2ED+E1-+75+-BC"


0
1
2
3
5
6
56/28
2

```
uint64_t get_value(uint64_t n) {  
    assert(n < N);  
    return n + 1;  
}
```

```
void put_value(uint64_t n, uint64_t v) {  
    assert(n < N);  
    assert(v == n + 4);  
}
```

Assignment #2

Przykład poprawnego obliczenia dla $N = 2$ (jako napis w języku C):

"01234n+P56789E-+D+*G*1n-+S2ED+E1-+75+-BC"



0
1
2
3
5
6
56/28
2
1

```
uint64_t get_value(uint64_t n) {  
    assert(n < N);  
    return n + 1;  
}
```

```
void put_value(uint64_t n, uint64_t v) {  
    assert(n < N);  
    assert(v == n + 4);  
}
```


Assignment #2

Przykład poprawnego obliczenia dla $N = 2$ (jako napis w języku C):

"01234n+P56789E-+D+*G*1n-+S2ED+E1-+75+-BC"



0
1
2
3
5
6
56/28
2
-1

```
uint64_t get_value(uint64_t n) {  
    assert(n < N);  
    return n + 1;  
}
```

```
void put_value(uint64_t n, uint64_t v) {  
    assert(n < N);  
    assert(v == n + 4);  
}
```

Assignment #2

Przykład poprawnego obliczenia dla $N = 2$ (jako napis w języku C):

"01234n+P56789E-+D+*G*1n-+S2ED+E1-+75+-BC"



0
1
2
3
5
6
56/28
1

```
uint64_t get_value(uint64_t n) {  
    assert(n < N);  
    return n + 1;  
}
```

```
void put_value(uint64_t n, uint64_t v) {  
    assert(n < N);  
    assert(v == n + 4);  
}
```

Assignment #2

Przykład poprawnego obliczenia dla $N = 2$ (jako napis w języku C):

"01234n+P56789E-+D+*G*1n-+S2ED+E1-+75+-BC"
↑

0
1
2
3
5
6
56/28
1
7
5

```
uint64_t get_value(uint64_t n) {  
    assert(n < N);  
    return n + 1;  
}
```

```
void put_value(uint64_t n, uint64_t v) {  
    assert(n < N);  
    assert(v == n + 4);  
}
```

Assignment #2

Przykład poprawnego obliczenia dla $N = 2$ (jako napis w języku C):

"01234n+P56789E-+D+*G*1n-+S2ED+E1-+75+-BC"
↑

0
1
2
3
5
6
56/28
1
12

```
uint64_t get_value(uint64_t n) {  
    assert(n < N);  
    return n + 1;  
}
```

```
void put_value(uint64_t n, uint64_t v) {  
    assert(n < N);  
    assert(v == n + 4);  
}
```

Assignment #2

Przykład poprawnego obliczenia dla $N = 2$ (jako napis w języku C):

"01234n+P56789E-+D+*G*1n-+S2ED+E1-+75+-BC"
↑

0
1
2
3
5
6
56/28
1
-12

```
uint64_t get_value(uint64_t n) {  
    assert(n < N);  
    return n + 1;  
}
```

```
void put_value(uint64_t n, uint64_t v) {  
    assert(n < N);  
    assert(v == n + 4);  
}
```

Assignment #2

Przykład poprawnego obliczenia dla $N = 2$ (jako napis w języku C):

"01234n+P56789E-+D+*G*1n-+S2ED+E1-+75+-BC"




0
1
2
3
5
6
56/28
1

```
uint64_t get_value(uint64_t n) {  
    assert(n < N);  
    return n + 1;  
}
```

```
void put_value(uint64_t n, uint64_t v) {  
    assert(n < N);  
    assert(v == n + 4);  
}
```

Assignment #2

Przykład poprawnego obliczenia dla $N = 2$ (jako napis w języku C):

"01234n+P56789E-+D+*G*1n-+S2ED+E1-+75+-BC"


0
1
2
3
5
6
1
56/28

```
uint64_t get_value(uint64_t n) {  
    assert(n < N);  
    return n + 1;  
}
```

```
void put_value(uint64_t n, uint64_t v) {  
    assert(n < N);  
    assert(v == n + 4);  
}
```

Assignment #2

Przykład poprawnego obliczenia dla $N = 2$ (jako napis w języku C):

"01234n+P56789E-+D+*G*1n-+S2ED+E1-+75+-BC"



0
1
2
3
5
6
1
56/28
56/28

```
uint64_t get_value(uint64_t n) {  
    assert(n < N);  
    return n + 1;  
}
```

```
void put_value(uint64_t n, uint64_t v) {  
    assert(n < N);  
    assert(v == n + 4);  
}
```


Assignment #2

Przykład poprawnego obliczenia dla $N = 2$ (jako napis w języku C):

"01234n+P56789E-+D+*G*1n-+S2ED+E1-+75+-BC"




0
1
2
3
5
6
1
112/56

```
uint64_t get_value(uint64_t n) {  
    assert(n < N);  
    return n + 1;  
}
```

```
void put_value(uint64_t n, uint64_t v) {  
    assert(n < N);  
    assert(v == n + 4);  
}
```

Assignment #2

Przykład poprawnego obliczenia dla $N = 2$ (jako napis w języku C):

"01234n+P56789E-+D+*G*1n-+S2ED+E1-+75+-BC"


0
1
2
3
5
6
112/56
1

```
uint64_t get_value(uint64_t n) {  
    assert(n < N);  
    return n + 1;  
}
```

```
void put_value(uint64_t n, uint64_t v) {  
    assert(n < N);  
    assert(v == n + 4);  
}
```

Assignment #2

Przykład poprawnego obliczenia dla $N = 2$ (jako napis w języku C):

"01234n+P56789E-+D+*G*1n-+S2ED+E1-+75+-BC"



0
1
2
3
5
6
112/56
1
1

```
uint64_t get_value(uint64_t n) {  
    assert(n < N);  
    return n + 1;  
}
```

```
void put_value(uint64_t n, uint64_t v) {  
    assert(n < N);  
    assert(v == n + 4);  
}
```

Assignment #2

Przykład poprawnego obliczenia dla $N = 2$ (jako napis w języku C):

"01234n+P56789E-+D+*G*1n-+S2ED+E1-+75+-BC"



0
1
2
3
5
6
112/56
1
-1

```
uint64_t get_value(uint64_t n) {  
    assert(n < N);  
    return n + 1;  
}
```

```
void put_value(uint64_t n, uint64_t v) {  
    assert(n < N);  
    assert(v == n + 4);  
}
```

Assignment #2

Przykład poprawnego obliczenia dla $N = 2$ (jako napis w języku C):

"01234n+P56789E-+D+*G*1n-+S2ED+E1-+75+-BC"




0
1
2
3
5
6
112/56
0

```
uint64_t get_value(uint64_t n) {  
    assert(n < N);  
    return n + 1;  
}
```

```
void put_value(uint64_t n, uint64_t v) {  
    assert(n < N);  
    assert(v == n + 4);  
}
```

Assignment #2

Przykład poprawnego obliczenia dla $N = 2$ (jako napis w języku C):

"01234n+P56789E-+D+*G*1n-+S2ED+E1-+75+-BC"


0
1
2
3
5
6
112/56
0
7
5

```
uint64_t get_value(uint64_t n) {  
    assert(n < N);  
    return n + 1;  
}
```

```
void put_value(uint64_t n, uint64_t v) {  
    assert(n < N);  
    assert(v == n + 4);  
}
```

Assignment #2

Przykład poprawnego obliczenia dla $N = 2$ (jako napis w języku C):

"01234n+P56789E-+D+*G*1n-+S2ED+E1-+75+-BC"



0
1
2
3
5
6
112/56
0
12

```
uint64_t get_value(uint64_t n) {  
    assert(n < N);  
    return n + 1;  
}
```

```
void put_value(uint64_t n, uint64_t v) {  
    assert(n < N);  
    assert(v == n + 4);  
}
```

Assignment #2

Przykład poprawnego obliczenia dla $N = 2$ (jako napis w języku C):

"01234n+P56789E-+D+*G*1n-+S2ED+E1-+75+-BC"



0
1
2
3
5
6
112/56
0
-12


```
uint64_t get_value(uint64_t n) {  
    assert(n < N);  
    return n + 1;  
}
```

```
void put_value(uint64_t n, uint64_t v) {  
    assert(n < N);  
    assert(v == n + 4);  
}
```


Assignment #2

Przykład poprawnego obliczenia dla $N = 2$ (jako napis w języku C):

"01234n+P56789E-+D+*G*1n-+S2ED+E1-+75+-BC"



0
1
2
3
5
6
112/56
0


```
uint64_t get_value(uint64_t n) {  
    assert(n < N);  
    return n + 1;  
}
```

```
void put_value(uint64_t n, uint64_t v) {  
    assert(n < N);  
    assert(v == n + 4);  
}
```

Assignment #2

Przykład poprawnego obliczenia dla $N = 2$ (jako napis w języku C):

"01234n+P56789E-+D+*G*1n-+S2ED+E1-+75+-BC\0"



0
1
2
3
5
6
112/56

```
uint64_t get_value(uint64_t n) {  
    assert(n < N);  
    return n + 1;  
}
```

```
void put_value(uint64_t n, uint64_t v) {  
    assert(n < N);  
    assert(v == n + 4);  
}
```

Assignment #2

```
global euron
extern get_value, put_value
```

```
section .bss          ← ; align
```

```
...
```

```
section .text        ← ; align
```

```
...
```

```
euron:
```

```
    push    rbp
```

```
    mov     rbp, rsp
```

```
...
```

```
    mov     dl, [rsi]
```

```
    cmp     dl, 'B'
```

```
...
```

```
    ret          ← ; clean up
```

Assignment #2

Makefile

```
euron: euronmain.c euron.o
    gcc -o euron euronmain.c euron.o

%.o: %.asm
    nasm -DN=XXX -f elf64 -F dwarf -g $<
```

Assignment #2

Expected in:

<https://svn.mimuw.edu.pl/repos/SO/studenci/<login>/zadanie2>

with a proper filename and no additional files

Expected by:

01 April 2019, 8 p.m.

Assignment **#1**

Expected in:

<https://svn.mimuw.edu.pl/repos/SO/studenci/<login>/zadanie1>

with a proper filename and no additional files

Expected by:

22 March 2019, 8 p.m.

Assignment #1

Make sure that:

- ★ uninitialized data are kept in .bss section
 - ★ input bytes are buffered
- ★ sequences like 6, 8, 6, 8, 0, 2, 0 are detected
- ★ no initial values of 'ordinary' registers are assumed
- ★ syscall does not overwrite important registers