In [1]:
```
import os
```

In [ ]:
```
# Define the function to run the fib.py on a specific core
def run_on_core(core_num):
    if core_num not in [0, 1]:
        print("Invalid core number. Choose either 0 or 1.")
        return
    os.system(f"taskset -c {core_num} python fib.py")

# Take user input
core_num = int(input("Enter core number (0 or 1): "))
run_on_core(core_num)
```

```
Enter core number (0 or 1): 0
```

In [ ]:
```

```

```
21      cout << "WES237A lab 3" << endl;
22
23      char key=0;
24
25      // 1 argument on command line: delay = arg
26      if(argc >= 2)
27      {
28          delay = atoi(argv[1]);
29      }
30
31      // Declare 2 cpu_count variables: 1 for before sleeping, 1 for after sleeping
32      unsigned int cpu_before, cpu_after;
33
34      // Initialize the counter
35      init_counters(1, 0);
36
37      // Get the cyclecount before sleeping
38      cpu_before = get_cyclecount();
39
40      // Sleep for the specified delay time
41      usleep(delay * 1000000);
42
43      // Get the cyclecount after sleeping
44      cpu_after = get_cyclecount();
45
46      // Subtract the before and after cyclecount
47      unsigned int cpu_diff = cpu_after - cpu_before;
48
49      // Print the cycle count
50      cout << "Cycle count difference: " << cpu_diff << endl;
51
52      LinuxTimer t;
53      usleep(delay * 1000000);
54      t.stop();
55      cpu_timer = t.getElapsed();
56
57      cout << "Timer: " << (double)cpu_timer/1000000000.0 << endl;
58
59      return 0;
60  }
61
```