

CS 150: Processes

Cynthia Taylor

Oberlin College

April 21~~st~~ 2014

5

CS Board Games Night

- Tonight, 7pm
- Math Library, 2nd floor of King
- Everyone is welcome

CS Community Art Show

- 8pm this Saturday
- Harkness Dining Room

Test 2

- If you did not get it back during lab, you need to come get it from my office

Your Computer Does Multiple Things At the Same Time

- Powerpoint, terminal, komodo, etc
- Can run lots of different programs at once

Processes

- A process is a set of sequential instructions being run by your computer
 - Essentially, a running program
- So far, all of your programs have been single processes
- This week that changes!

Why use multiple processes?

- Want to get work done while one part of the program is waiting for something (user input, network, etc)

input()

- Want to take advantage of your computer's full power to get work done faster

We can programmatically create and run new processes

- Use the python multiprogramming module
- Create a function that does whatever we want our process to do
- Create a new process to run that function

Processes

```
def toRun(str, num):  
    for i in range(num):  
        print(str)
```

function

```
p = Process(target=toRun,  
args=("hi", 12))  
p.start()
```

str num

Some Weird Stuff

- Process takes named arguments (target, args)

```
p = Process(target=toRun, args=("hi", 12))
```

- args is a tuple
 - Even with one argument, need a comma: (x,)
- We're passing a function as an argument

To create a new process to run this procedure on arguments d,e,f

```
def newProc(a,b,c):  
    ...
```

A. `p = Process(newProc,d,e,f)`

B. `p = Process(target=newProc,
args=(d,e,f))`

C. `p = Process(target=newProc,
args=(d,e,f))`

D. `p = Process(target=newProc(a,b,c),
args=(d,e,f))`

E. I don't know

no named
args

tuple

Process ID

- Every process has a unique process id number, or pid, assigned by the operating system.
- To access the pid for a process, use current_process().pid

Multiprocess Demo

Process Scheduling

- Note that the computer controls when to run different processes, and when to switch between them
- We cannot control when we switch between processes, or the order which things in different processes happen

We have two processes, A and B. Both have an execution time of 20 ms. A starts before B. Which will finish running first?

A. A

B. B

C. There is no way to tell

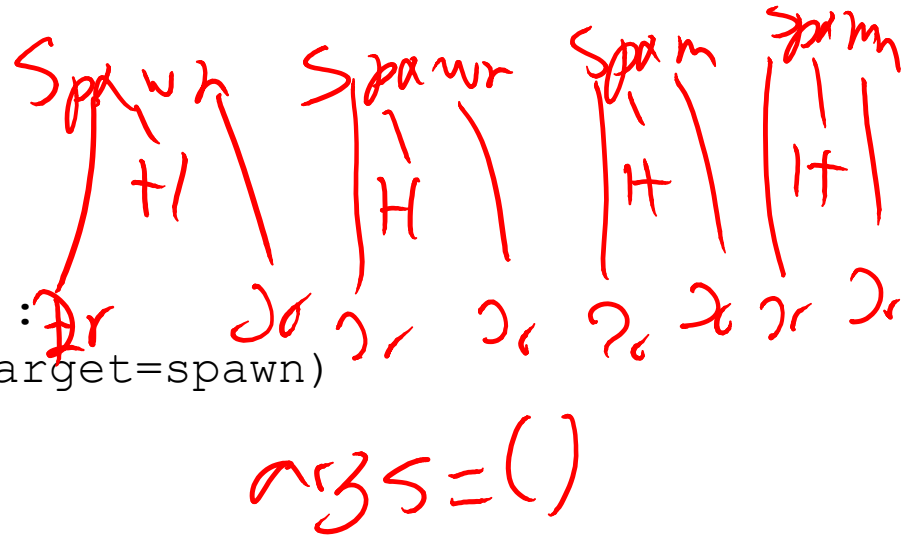
D. I don't know

How many times will “Hello” print?

```
def spawn():  
    print("Hello")  
    for i in range(2):  
        p = Process(target=toRun)  
        p.start()
```

```
def toRun():  
    print("Hello")
```

```
def main():  
    for i in range(4):  
        p = Process(target=spawn)  
        p.start()
```



A. 2 B. 8 **C. 12** D. 16

E. I don't know

Spawning Processes Demo

Next Time

- Multiple Processes – Process Pools
- Lab 10 – Tuesday at 10 pm
- Prelab 11 – Wednesday in class