

\* Key \*



THE UNIVERSITY of  
NEW ORLEANS

DEPARTMENT OF  
COMPUTER SCIENCE

CSCI 2467, Fall 2019

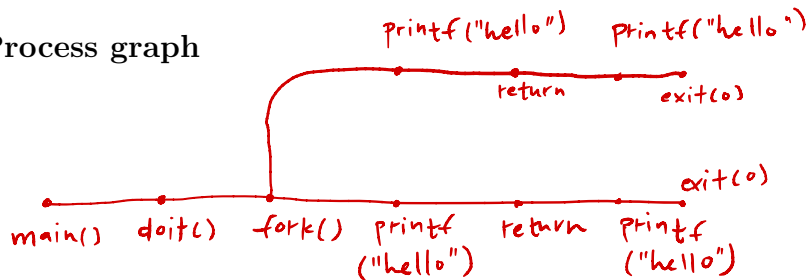
**Class Activity:** graphing and predicting a forking program  
Tuesday, October 29

## 1 Forecasting a fork()

Consider the program below, which contains a `main()` function which calls another function called `doit()`. The C source for this program is also available from the 2467 schedule page, activity link.

```
void doit () {  
    fork ();  
    printf ("hello\n");  
    return;  
}  
  
int main () {  
    doit ();  
    printf ("hello\n");  
    exit (0);  
}
```

### 1.1 Process graph



In the space above, draw a process graph, similar to the ones seen in the text and in the lecture slides. Each invocation of `doit()`, `fork()`, and `printf()` should be a node on the graph.

### 1.2 Output

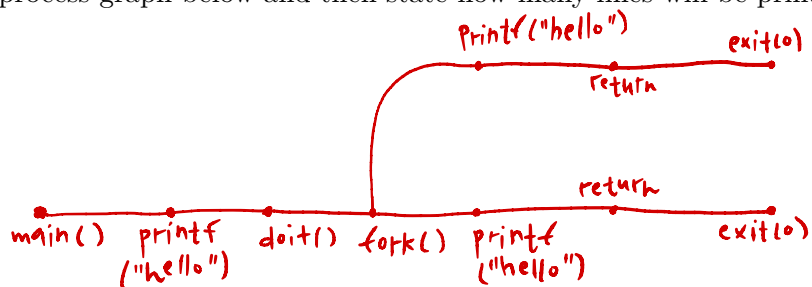
How many lines of output (i.e. how many instances of “hello”) does this program produce?

4

### 1.3 Change order of main()

In `main()`, if you put the `printf()` statement *before* `doit()`, how does this change the output?

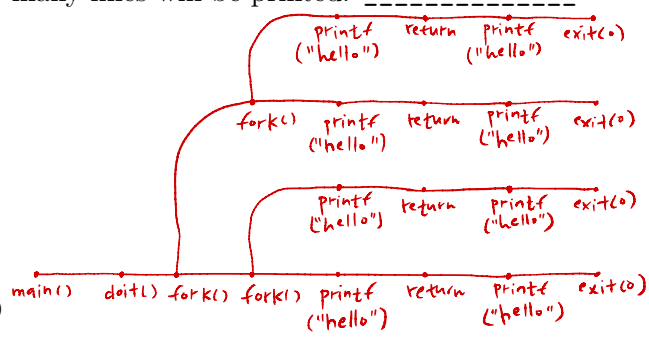
Draw a new process graph below and then state how many lines will be printed: 3



### 1.4 Add another fork() to doit()

In `doit()`, if you add a second call to `fork()` at the beginning of the function, how does this change the output?

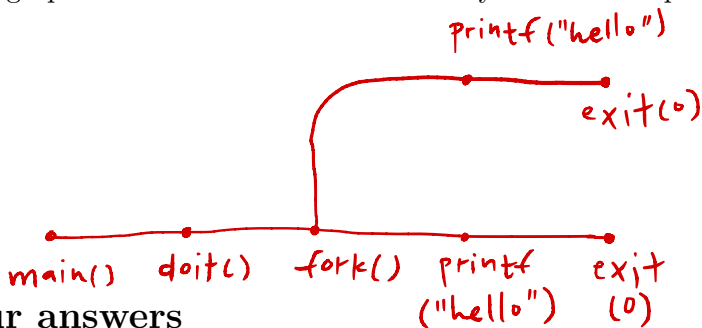
Draw a new process graph below and then state how many lines will be printed: 8



### 1.5 Replace return with exit(0) in doit()

In `doit()`, if you replace `return` with `exit(0)`, how does this change the output?

Draw a new process graph below and then state how many lines will be printed: 2



## 2 Check your answers

Now that you have 3 graphs and 3 answers for the problem above, make a new directory and then create 3 files with the C source for these three examples: `doit1.c`, `doit2.c`, `doit3.c`.

After you create each file and make changes to the source code, you can compile each by running `make doit1` and so on. (This is simply a shortcut to running the gcc compiler)

Run each program and compare to your process graph and prediction. How did the output of those programs compare to what you predicted?