

Princeton University
COS 217: Introduction to Programming Systems
Trace of testforkexit

```
% gcc217 testforkexit.c -o testforkexit
```

Princeton University
COS 217: Introduction to Programming Systems
Trace of testforkexit

```
% ./testforkexit
```

```
29595
```

```
int main(void)
{
    pid_t iPid;
    printf("%d parent\n",
        (int)getPid());
    fflush(stdin);
    fflush(stdout);
    iPid = fork();
    if (iPid == 0)
    {
        printf("%d child\n",
            (int)getPid());
        exit(0);
    }
    printf("%d parent\n",
        (int)getPid());
    return 0;
}
```

Princeton University
COS 217: Introduction to Programming Systems
Trace of testforkexit

```
% ./testforkexit
```

```
29595
```

```
int main(void)
{
    pid_t iPid;
    printf("%d parent\n",
           (int)getPid());
    fflush(stdin);
    fflush(stdout);
    iPid = fork();
    if (iPid == 0)
    {
        printf("%d child\n",
               (int)getPid());
        exit(0);
    }
    printf("%d parent\n",
           (int)getPid());
    return 0;
}
```

Princeton University
COS 217: Introduction to Programming Systems
Trace of testforkexit

% ./testforkexit

29595

```
int main(void)
{
    pid_t iPid;
    printf("%d parent\n",
           (int)getPid());
    fflush(stdin);
    fflush(stdout);
    iPid = fork();
    if (iPid == 0)
    {
        printf("%d child\n",
               (int)getPid());
        exit(0);
    }
    printf("%d parent\n",
           (int)getPid());
    return 0;
}
```

Writes:

29595 parent

Princeton University
COS 217: Introduction to Programming Systems
Trace of testforkexit

```
% ./testforkexit
```

```
29595
```

```
int main(void)
{
    pid_t iPid;
    printf("%d parent\n",
           (int)getPid());
    fflush(stdin);
    fflush(stdout);
    iPid = fork();
    if (iPid == 0)
    {
        printf("%d child\n",
               (int)getPid());
        exit(0);
    }
    printf("%d parent\n",
           (int)getPid());
    return 0;
}
```

Princeton University
COS 217: Introduction to Programming Systems
Trace of testforkexit

```
% ./testforkexit
```

```
29595
```

```
int main(void)
{
    pid_t iPid;
    printf("%d parent\n",
        (int)getPid());
    fflush(stdin);
    fflush(stdout);
    iPid = fork();
    if (iPid == 0)
    {
        printf("%d child\n",
            (int)getPid());
        exit(0);
    }
    printf("%d parent\n",
        (int)getPid());
    return 0;
}
```

Princeton University
COS 217: Introduction to Programming Systems
Trace of testforkexit

```
% ./testforkexit
```

```
29595
int main(void)
{
    pid_t iPid;
    printf("%d parent\n",
           (int)getPid());
    fflush(stdin);
    fflush(stdout);
    iPid = fork();
    if (iPid == 0)
    {
        printf("%d child\n",
               (int)getPid());
        exit(0);
    }
    printf("%d parent\n",
           (int)getPid());
    return 0;
}
```

concurrent

```
29596
int main(void)
{
    pid_t iPid;
    printf("%d parent\n",
           (int)getPid());
    fflush(stdin);
    fflush(stdout);
    iPid = fork();
    if (iPid == 0)
    {
        printf("%d child\n",
               (int)getPid());
        exit(0);
    }
    printf("%d parent\n",
           (int)getPid());
    return 0;
}
```

Assume OS gives CPU to parent

Princeton University
COS 217: Introduction to Programming Systems
Trace of testforkexit

```
% ./testforkexit
```

```
29595
int main(void)
{
    pid_t iPid;
    printf("%d parent\n",
        (int)getPid());
    fflush(stdin);
    fflush(stdout);
    iPid = fork();
    if (iPid == 0)
    {
        printf("%d child\n",
            (int)getPid());
        exit(0);
    }
    printf("%d parent\n",
        (int)getPid());
    return 0;
}
```

29596

```
29596
int main(void)
{
    pid_t iPid;
    printf("%d parent\n",
        (int)getPid());
    fflush(stdin);
    fflush(stdout);
    iPid = fork();
    if (iPid == 0)
    {
        printf("%d child\n",
            (int)getPid());
        exit(0);
    }
    printf("%d parent\n",
        (int)getPid());
    return 0;
}
```

concurrent

Assume OS gives CPU to parent

Princeton University
COS 217: Introduction to Programming Systems
Trace of testforkexit

```
% ./testforkexit
```

```
29595
int main(void)
{
    pid_t iPid;
    printf("%d parent\n",
           (int)getPid());
    fflush(stdin);
    fflush(stdout);
    iPid = fork();
    if (iPid == 0)
    {
        printf("%d child\n",
               (int)getPid());
        exit(0);
    }
    printf("%d parent\n",
           (int)getPid());
    return 0;
}
```

29596

```
29596
int main(void)
{
    pid_t iPid;
    printf("%d parent\n",
           (int)getPid());
    fflush(stdin);
    fflush(stdout);
    iPid = fork();
    if (iPid == 0)
    {
        printf("%d child\n",
               (int)getPid());
        exit(0);
    }
    printf("%d parent\n",
           (int)getPid());
    return 0;
}
```

concurrent

Assume OS gives CPU to parent

Princeton University
COS 217: Introduction to Programming Systems
Trace of testforkexit

```
% ./testforkexit
```

```
29595
int main(void)
{
    pid_t iPid;
    printf("%d parent\n",
           (int)getPid());
    fflush(stdin);
    fflush(stdout);
    iPid = fork();
    if (iPid == 0)
    {
        printf("%d child\n",
               (int)getPid());
        exit(0);
    }
    printf("%d parent\n",
           (int)getPid());
    return 0;
}
```

29596

```
29596
int main(void)
{
    pid_t iPid;
    printf("%d parent\n",
           (int)getPid());
    fflush(stdin);
    fflush(stdout);
    iPid = fork();
    if (iPid == 0)
    {
        printf("%d child\n",
               (int)getPid());
        exit(0);
    }
    printf("%d parent\n",
           (int)getPid());
    return 0;
}
```

concurrent

Writes:

29595 parent

Assume OS gives CPU to parent

Princeton University
COS 217: Introduction to Programming Systems
Trace of testforkexit

% ./testforkexit

```
29595
int main(void)
{
    pid_t iPid;
    printf("%d parent\n",
           (int) getpid());
    fflush(stdin);
    fflush(stdout);
    iPid = fork();
    if (iPid == 0)
    {
        printf("%d child\n",
               (int) getpid());
        exit(0);
    }
    printf("%d parent\n",
           (int) getpid());
    return 0;
}
```

29596

concurrent

```
29596
int main(void)
{
    pid_t iPid;
    printf("%d parent\n",
           (int) getpid());
    fflush(stdin);
    fflush(stdout);
    iPid = fork();
    if (iPid == 0)
    {
        printf("%d child\n",
               (int) getpid());
        exit(0);
    }
    printf("%d parent\n",
           (int) getpid());
    return 0;
}
```

Princeton University
COS 217: Introduction to Programming Systems
Trace of testforkexit

```
% ./testforkexit
```

```
29596
```

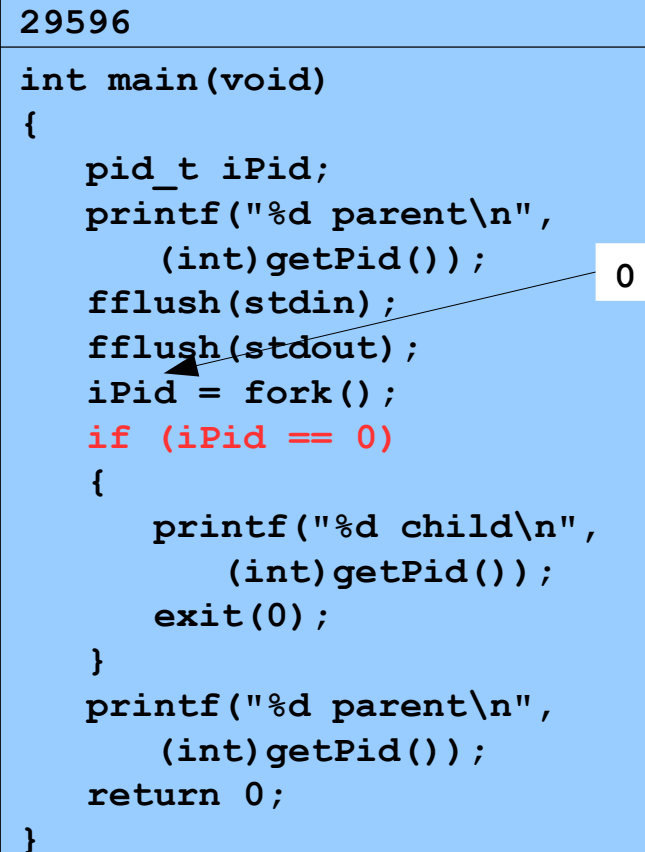
```
int main(void)
{
    pid_t iPid;
    printf("%d parent\n",
           (int) getpid());
    fflush(stdin);
    fflush(stdout);
    iPid = fork();
    if (iPid == 0)
    {
        printf("%d child\n",
               (int) getpid());
        exit(0);
    }
    printf("%d parent\n",
           (int) getpid());
    return 0;
}
```

```
0
```

Princeton University
COS 217: Introduction to Programming Systems
Trace of testforkexit

```
% ./testforkexit
```

```
29596
int main(void)
{
    pid_t iPid;
    printf("%d parent\n",
           (int) getpid());
    fflush(stdin);
    fflush(stdout);
    iPid = fork();
    if (iPid == 0)
    {
        printf("%d child\n",
               (int) getpid());
        exit(0);
    }
    printf("%d parent\n",
           (int) getpid());
    return 0;
}
```



Princeton University
COS 217: Introduction to Programming Systems
Trace of testforkexit

```
% ./testforkexit
```

```
29596
int main(void)
{
    pid_t iPid;
    printf("%d parent\n",
           (int) getpid());
    fflush(stdin);
    fflush(stdout);
    iPid = fork();
    if (iPid == 0)
    {
        printf("%d child\n",
               (int) getpid());
        exit(0);
    }
    printf("%d parent\n",
           (int) getpid());
    return 0;
}
```

0

Writes:
29596 child

Princeton University
COS 217: Introduction to Programming Systems
Trace of testforkexit

```
% ./testforkexit
```

```
29596
int main(void)
{
    pid_t iPid;
    printf("%d parent\n",
           (int) getpid());
    fflush(stdin);
    fflush(stdout);
    iPid = fork();
    if (iPid == 0)
    {
        printf("%d child\n",
               (int) getpid());
        exit(0);
    }
    printf("%d parent\n",
           (int) getpid());
    return 0;
}
```

0

Writes:
29596 child

Princeton University
COS 217: Introduction to Programming Systems
Trace of testforkexit

⊘

Copyright © 2019 by Robert M. Dondero, Jr.