

**Princeton University**  
**COS 217: Introduction to Programming Systems**  
**Trace of testforkloop**

```
% gcc217 testforkloop.c -o testforkloop
```

**Princeton University**  
**COS 217: Introduction to Programming Systems**  
**Trace of testforkloop**

```
% ./testforkloop
```

```
9857
```

```
int main(void)
{
    pid_t iPid;
    int i;
    printf("%d parent\n",
           (int)getPid());
    fflush(stdin);
    fflush(stdout);
    iPid = fork();
    if (iPid == 0)
    { for (i = 0; i < 10; i++)
      printf("%d child %d\n",
             (int)getPid(), i);
      exit(0);
    }
    for (i = 0; i < 10; i++)
        printf("%d parent %d\n",
               (int)getPid(), i);
    return 0;
}
```

**Princeton University**  
**COS 217: Introduction to Programming Systems**  
**Trace of testforkloop**

```
% ./testforkloop
```

```
9857
```

```
int main(void)
{
    pid_t iPid;
    int i;
    printf("%d parent\n",
           (int)getPid());
    fflush(stdin);
    fflush(stdout);
    iPid = fork();
    if (iPid == 0)
    { for (i = 0; i < 10; i++)
        printf("%d child %d\n",
               (int)getPid(), i);
        exit(0);
    }
    for (i = 0; i < 10; i++)
        printf("%d parent %d\n",
               (int)getPid(), i);
    return 0;
}
```

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

```
% ./testforkloop
```

```
9857
```

```
int main(void)
{
    pid_t iPid;
    int i;
    printf("%d parent\n",
           (int)getPid());
    fflush(stdin);
    fflush(stdout);
    iPid = fork();
    if (iPid == 0)
    { for (i = 0; i < 10; i++)
      printf("%d child %d\n",
             (int)getPid(), i);
      exit(0);
    }
    for (i = 0; i < 10; i++)
        printf("%d parent %d\n",
               (int)getPid(), i);
    return 0;
}
```

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

```
% ./testforkloop
```

```
9857
```

```
int main(void)
{
    pid_t iPid;
    int i;
    printf("%d parent\n",
           (int)getPid());
    fflush(stdin);
    fflush(stdout);
    iPid = fork();
    if (iPid == 0)
    { for (i = 0; i < 10; i++)
      printf("%d child %d\n",
             (int)getPid(), i);
      exit(0);
    }
    for (i = 0; i < 10; i++)
        printf("%d parent %d\n",
               (int)getPid(), i);
    return 0;
}
```

Writes:

```
9857 parent
```

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

```
% ./testforkloop
```

```
9857
```

```
int main(void)
{
    pid_t iPid;
    int i;
    printf("%d parent\n",
        (int)getPid());
    fflush(stdin);
    fflush(stdout);
    iPid = fork();
    if (iPid == 0)
    {
        for (i = 0; i < 10; i++)
            printf("%d child %d\n",
                (int)getPid(), i);
        exit(0);
    }
    for (i = 0; i < 10; i++)
        printf("%d parent %d\n",
            (int)getPid(), i);
    return 0;
}
```

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

```
% ./testforkloop
```

```
9857
```

```
int main(void)
{
    pid_t iPid;
    int i;
    printf("%d parent\n",
        (int)getPid());
    fflush(stdin);
    fflush(stdout);
    iPid = fork();
    if (iPid == 0)
    { for (i = 0; i < 10; i++)
        printf("%d child %d\n",
            (int)getPid(), i);
        exit(0);
    }
    for (i = 0; i < 10; i++)
        printf("%d parent %d\n",
            (int)getPid(), i);
    return 0;
}
```

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

```
% ./testforkloop
```

```
9857
int main(void)
{
    pid_t iPid;
    int i;
    printf("%d parent\n",
           (int)getPid());
    fflush(stdin);
    fflush(stdout);
    iPid = fork();
    if (iPid == 0)
    { for (i = 0; i < 10; i++)
      printf("%d child %d\n",
             (int)getPid(), i);
      exit(0);
    }
    for (i = 0; i < 10; i++)
        printf("%d parent %d\n",
               (int)getPid(), i);
    return 0;
}
```

concurrent

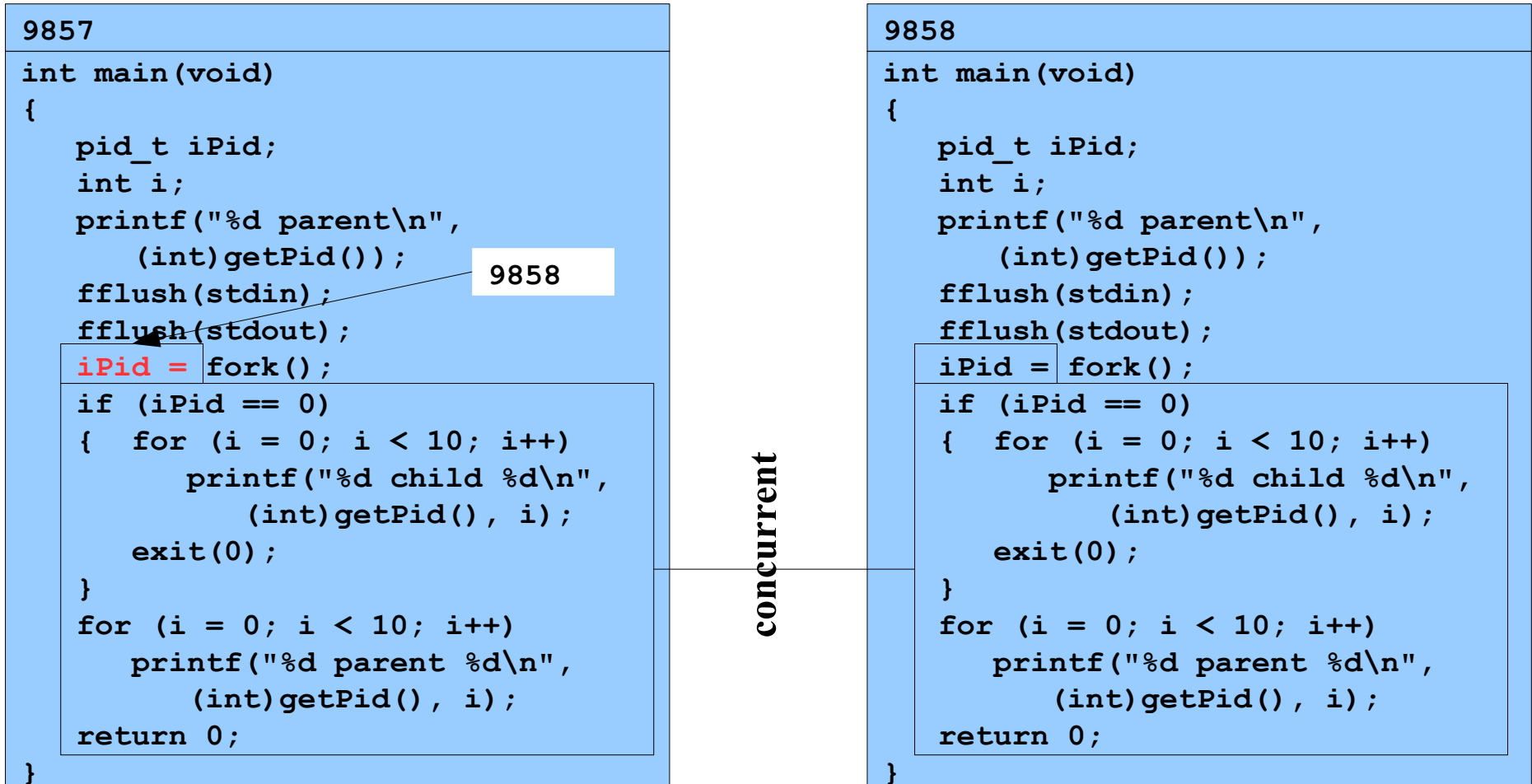
```
9858
int main(void)
{
    pid_t iPid;
    int i;
    printf("%d parent\n",
           (int)getPid());
    fflush(stdin);
    fflush(stdout);
    iPid = fork();
    if (iPid == 0)
    { for (i = 0; i < 10; i++)
      printf("%d child %d\n",
             (int)getPid(), i);
      exit(0);
    }
    for (i = 0; i < 10; i++)
        printf("%d parent %d\n",
               (int)getPid(), i);
    return 0;
}
```

Assume OS gives CPU to parent



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

% ./testforkloop



Assume OS gives CPU to parent

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

% ./testforkloop

```
9857
int main(void)
{
    pid_t iPid;
    int i;
    printf("%d parent\n",
           (int)getPid());
    fflush(stdin);
    fflush(stdout);
    iPid = fork();
    if (iPid == 0)
    { for (i = 0; i < 10; i++)
      printf("%d child %d\n",
             (int)getPid(), i);
      exit(0);
    }
    for (i = 0; i < 10; i++)
        printf("%d parent %d\n",
               (int)getPid(), i);
    return 0;
}
```

9858

```
9858
int main(void)
{
    pid_t iPid;
    int i;
    printf("%d parent\n",
           (int)getPid());
    fflush(stdin);
    fflush(stdout);
    iPid = fork();
    if (iPid == 0)
    { for (i = 0; i < 10; i++)
      printf("%d child %d\n",
             (int)getPid(), i);
      exit(0);
    }
    for (i = 0; i < 10; i++)
        printf("%d parent %d\n",
               (int)getPid(), i);
    return 0;
}
```

concurrent

Assume OS gives CPU to parent

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

% ./testforkloop

```
9857
int main(void)
{
    pid_t iPid;
    int i;
    printf("%d parent\n",
           (int)getPid());
    fflush(stdin);
    fflush(stdout);
    iPid = fork();
    if (iPid == 0)
    { for (i = 0; i < 10; i++)
      printf("%d child %d\n",
             (int)getPid(), i);
      exit(0);
    }
    for (i = 0; i < 10; i++)
        printf("%d parent %d\n",
               (int)getPid(), i);
    return 0;
}
```

```
9858
int main(void)
{
    pid_t iPid;
    int i;
    printf("%d parent\n",
           (int)getPid());
    fflush(stdin);
    fflush(stdout);
    iPid = fork();
    if (iPid == 0)
    { for (i = 0; i < 10; i++)
      printf("%d child %d\n",
             (int)getPid(), i);
      exit(0);
    }
    for (i = 0; i < 10; i++)
        printf("%d parent %d\n",
               (int)getPid(), i);
    return 0;
}
```

concurrent

Writes:

9857 parent 0 ... 7

Assume OS gives CPU to parent

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

% ./testforkloop

```
9857
int main(void)
{
    pid_t iPid;
    int i;
    printf("%d parent\n",
           (int)getPid());
    fflush(stdin);
    fflush(stdout);
    iPid = fork();
    if (iPid == 0)
    { for (i = 0; i < 10; i++)
      printf("%d child %d\n",
             (int)getPid(), i);
      exit(0);
    }
    for (i = 0; i < 10; i++)
        printf("%d parent %d\n",
               (int)getPid(), i);
    return 0;
}
```

```
9858
int main(void)
{
    pid_t iPid;
    int i;
    printf("%d parent\n",
           (int)getPid());
    fflush(stdin);
    fflush(stdout);
    iPid = fork();
    if (iPid == 0)
    { for (i = 0; i < 10; i++)
      printf("%d child %d\n",
             (int)getPid(), i);
      exit(0);
    }
    for (i = 0; i < 10; i++)
        printf("%d parent %d\n",
               (int)getPid(), i);
    return 0;
}
```

concurrent

Assume OS gives CPU to child

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

% ./testforkloop

```
9857
int main(void)
{
    pid_t iPid;
    int i;
    printf("%d parent\n",
           (int)getPid());
    fflush(stdin);
    fflush(stdout);
    iPid = fork();
    if (iPid == 0)
    { for (i = 0; i < 10; i++)
      printf("%d child %d\n",
             (int)getPid(), i);
      exit(0);
    }
    for (i = 0; i < 10; i++)
        printf("%d parent %d\n",
               (int)getPid(), i);
    return 0;
}
```

```
9858
int main(void)
{
    pid_t iPid;
    int i;
    printf("%d parent\n",
           (int)getPid());
    fflush(stdin);
    fflush(stdout);
    iPid = fork();
    if (iPid == 0)
    { for (i = 0; i < 10; i++)
      printf("%d child %d\n",
             (int)getPid(), i);
      exit(0);
    }
    for (i = 0; i < 10; i++)
        printf("%d parent %d\n",
               (int)getPid(), i);
    return 0;
}
```

concurrent

Assume OS gives CPU to child

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

% ./testforkloop

```
9857
int main(void)
{
    pid_t iPid;
    int i;
    printf("%d parent\n",
           (int)getPid());
    fflush(stdin);
    fflush(stdout);
    iPid = fork();
    if (iPid == 0)
    { for (i = 0; i < 10; i++)
      printf("%d child %d\n",
             (int)getPid(), i);
      exit(0);
    }
    for (i = 0; i < 10; i++)
        printf("%d parent %d\n",
               (int)getPid(), i);
    return 0;
}
```

```
9858
int main(void)
{
    pid_t iPid;
    int i;
    printf("%d parent\n",
           (int)getPid());
    fflush(stdin);
    fflush(stdout);
    iPid = fork();
    if (iPid == 0)
    { for (i = 0; i < 10; i++)
      printf("%d child %d\n",
             (int)getPid(), i);
      exit(0);
    }
    for (i = 0; i < 10; i++)
        printf("%d parent %d\n",
               (int)getPid(), i);
    return 0;
}
```

concurrent

Writes:  
9858 child 0

Assume OS gives CPU to parent

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

% ./testforkloop

```
9857
int main(void)
{
    pid_t iPid;
    int i;
    printf("%d parent\n",
           (int)getPid());
    fflush(stdin);
    fflush(stdout);
    iPid = fork();
    if (iPid == 0)
    { for (i = 0; i < 10; i++)
      printf("%d child %d\n",
             (int)getPid(), i);
      exit(0);
    }
    for (i = 0; i < 10; i++)
        printf("%d parent %d\n",
               (int)getPid(), i);
    return 0;
}
```

```
9858
int main(void)
{
    pid_t iPid;
    int i;
    printf("%d parent\n",
           (int)getPid());
    fflush(stdin);
    fflush(stdout);
    iPid = fork();
    if (iPid == 0)
    { for (i = 0; i < 10; i++)
      printf("%d child %d\n",
             (int)getPid(), i);
      exit(0);
    }
    for (i = 0; i < 10; i++)
        printf("%d parent %d\n",
               (int)getPid(), i);
    return 0;
}
```

concurrent

Writes:

9857 parent 8

Assume OS gives CPU to child

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

% ./testforkloop

```
9857
int main(void)
{
    pid_t iPid;
    int i;
    printf("%d parent\n",
           (int)getPid());
    fflush(stdin);
    fflush(stdout);
    iPid = fork();
    if (iPid == 0)
    { for (i = 0; i < 10; i++)
      printf("%d child %d\n",
             (int)getPid(), i);
      exit(0);
    }
    for (i = 0; i < 10; i++)
        printf("%d parent %d\n",
               (int)getPid(), i);
    return 0;
}
```

```
9858
int main(void)
{
    pid_t iPid;
    int i;
    printf("%d parent\n",
           (int)getPid());
    fflush(stdin);
    fflush(stdout);
    iPid = fork();
    if (iPid == 0)
    { for (i = 0; i < 10; i++)
      printf("%d child %d\n",
             (int)getPid(), i);
      exit(0);
    }
    for (i = 0; i < 10; i++)
        printf("%d parent %d\n",
               (int)getPid(), i);
    return 0;
}
```

concurrent

Writes:  
9858 child 1

16

Assume OS gives CPU to parent



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

% ./testforkloop

```
9857
int main(void)
{
    pid_t iPid;
    int i;
    printf("%d parent\n",
           (int)getPid());
    fflush(stdin);
    fflush(stdout);
    iPid = fork();
    if (iPid == 0)
    { for (i = 0; i < 10; i++)
      printf("%d child %d\n",
             (int)getPid(), i);
      exit(0);
    }
    for (i = 0; i < 10; i++)
        printf("%d parent %d\n",
               (int)getPid(), i);
    return 0;
}
```

```
9858
int main(void)
{
    pid_t iPid;
    int i;
    printf("%d parent\n",
           (int)getPid());
    fflush(stdin);
    fflush(stdout);
    iPid = fork();
    if (iPid == 0)
    { for (i = 0; i < 10; i++)
      printf("%d child %d\n",
             (int)getPid(), i);
      exit(0);
    }
    for (i = 0; i < 10; i++)
        printf("%d parent %d\n",
               (int)getPid(), i);
    return 0;
}
```

concurrent

Writes:

9857 parent 9

Assume OS gives CPU to parent

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

% ./testforkloop

```
9857
int main(void)
{
    pid_t iPid;
    int i;
    printf("%d parent\n",
           (int)getPid());
    fflush(stdin);
    fflush(stdout);
    iPid = fork();
    if (iPid == 0)
    { for (i = 0; i < 10; i++)
      printf("%d child %d\n",
             (int)getPid(), i);
      exit(0);
    }
    for (i = 0; i < 10; i++)
    printf("%d parent %d\n",
           (int)getPid(), i);
    return 0;
}
```

9858

```
9858
int main(void)
{
    pid_t iPid;
    int i;
    printf("%d parent\n",
           (int)getPid());
    fflush(stdin);
    fflush(stdout);
    iPid = fork();
    if (iPid == 0)
    { for (i = 0; i < 10; i++)
      printf("%d child %d\n",
             (int)getPid(), i);
      exit(0);
    }
    for (i = 0; i < 10; i++)
    printf("%d parent %d\n",
           (int)getPid(), i);
    return 0;
}
```

0

concurrent

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

% ./testforkloop

```
9858
int main(void)
{
    pid_t iPid;
    int i;
    printf("%d parent\n",
           (int)getPid());
    fflush(stdin);
    fflush(stdout);
    iPid = fork();
    if (iPid == 0)
    { for (i = 0; i < 10; i++)
      printf("%d child %d\n",
             (int)getPid(), i);
      exit(0);
    }
    for (i = 0; i < 10; i++)
        printf("%d parent %d\n",
               (int)getPid(), i);
    return 0;
}
```

Writes:

9858 child 2 ... 9

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

% ./testforkloop

```
9858
int main(void)
{
    pid_t iPid;
    int i;
    printf("%d parent\n",
           (int)getPid());
    fflush(stdin);
    fflush(stdout);
    iPid = fork();
    if (iPid == 0)
    { for (i = 0; i < 10; i++)
      printf("%d child %d\n",
             (int)getPid(), i);
      exit(0);
    }
    for (i = 0; i < 10; i++)
    printf("%d parent %d\n",
           (int)getPid(), i);
    return 0;
}
```

**Princeton University**  
**COS 217: Introduction to Programming Systems**  
**Trace of testforkloop**

8