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

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

```
% ./testforkexit
```

```
29595

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

```
% ./testforkexit
```

```
29595

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

```
% ./testforkexit
```

```
29595

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

Writes:
```
29595 parent
```

```
% ./testforkexit
```

```
29595

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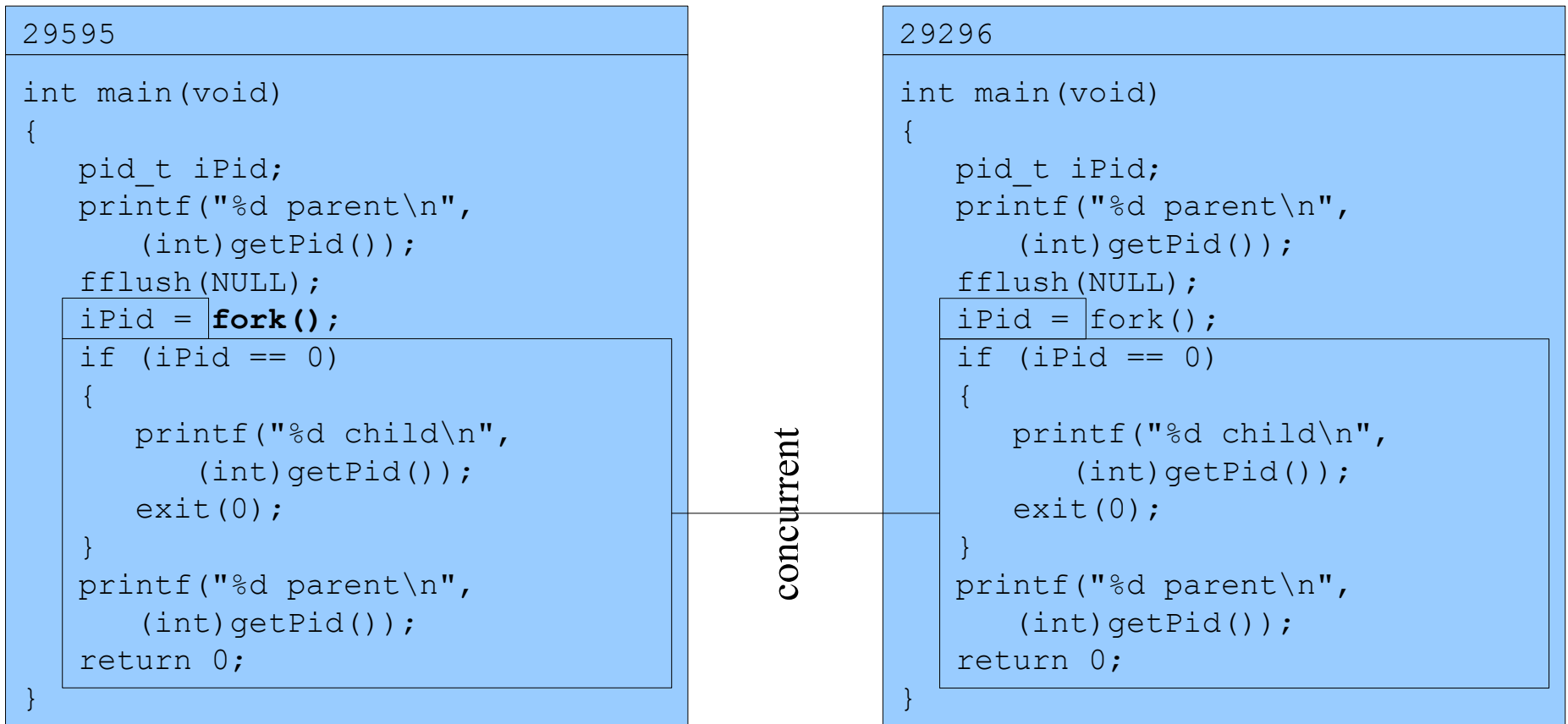
```
% ./testforkexit
```

29595

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

concurrent

29296

```
int main(void)
{
   pid_t iPid;
   printf("%d parent\n",
      (int)getPid());
   fflush(NULL);
   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

```
% ./testforkexit
```

29595

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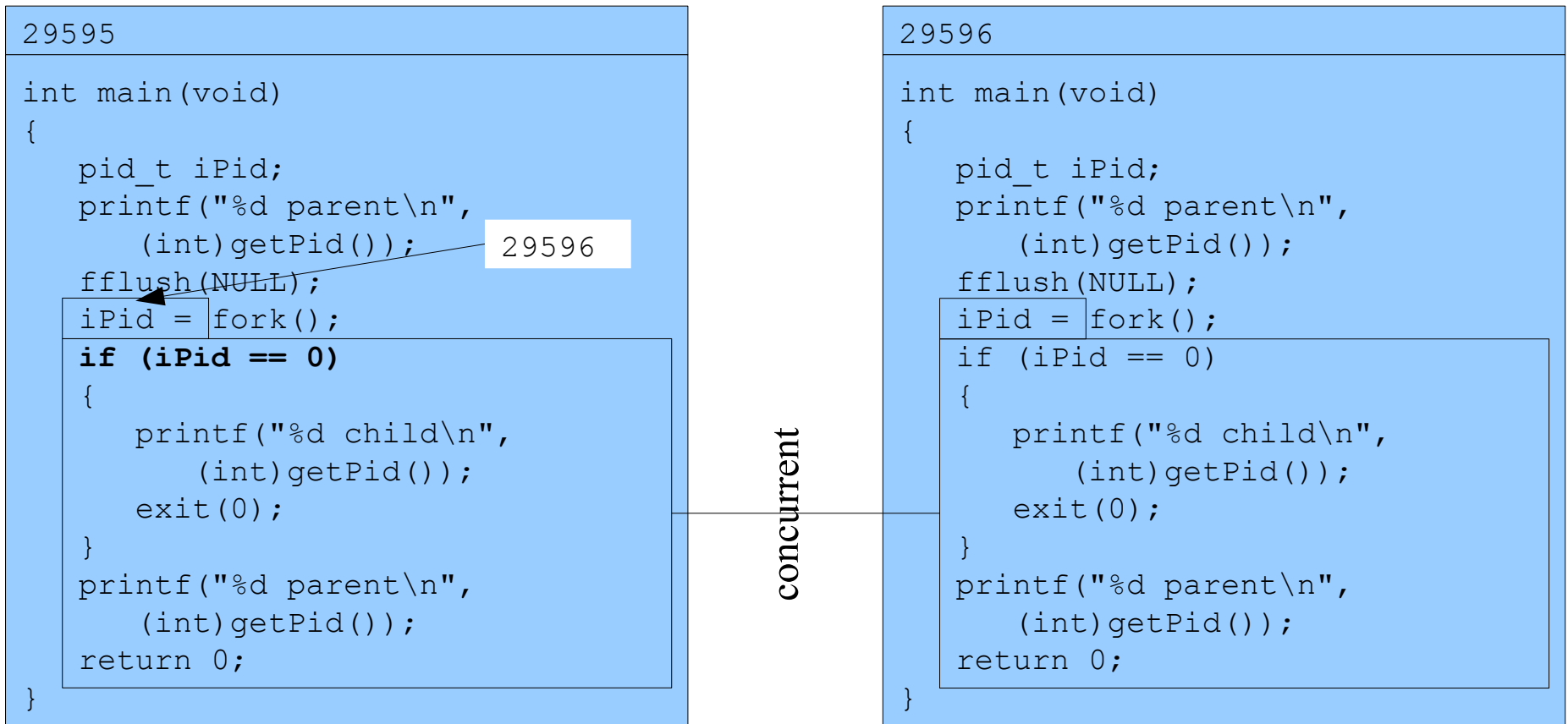
29596

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

29596

concurrent

Assume OS gives CPU to parent

7

```
% ./testforkexit
```

29595

```
int main(void)
{
   pid_t iPid;
   printf("%d parent\n",
      (int)getPid());
   fflush(NULL);
   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(NULL);
   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

8

```
% ./testforkexit
```

29595

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

29596

```
int main(void)
{
   pid_t iPid;
   printf("%d parent\n",
      (int)getPid());
   fflush(NULL);
   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

9

```
% ./testforkexit
```

29595

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

concurrent

```
% ./testforkexit
```

```
29596

int main(void)
{
   pid_t iPid;
   printf("%d parent\n",
      (int)getPid());        0
   fflush(NULL);
   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
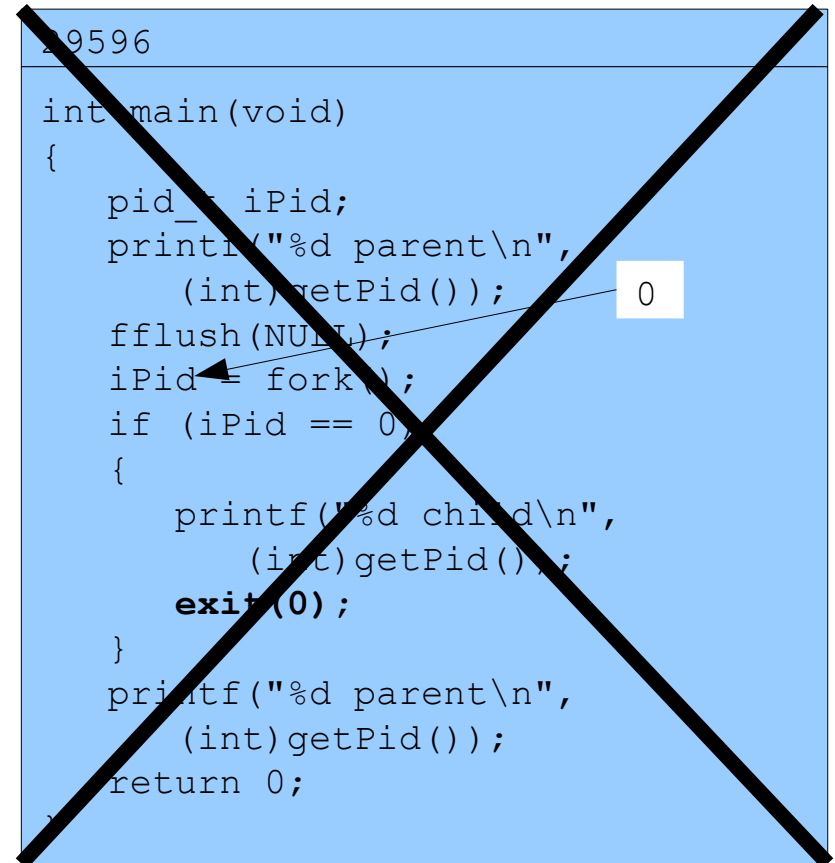
```
29596

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

```
% ./testforkexit
```

```
29596

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

Writes:
```
29596 child
```

```
% ./testforkexit
```

29596

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

0

%