```
% gcc217 testforkwait.c -o testforkwait
```

```
%  ./testforkwait
```

```
7275
```

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

```
% ./testforkwait
```

```
7275
```

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

```
% ./testforkwait
```

```
7275
```

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

```
% ./testforkwait
```

```
7275

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

Writes:
7275 parent

5

```
% ./testforkwait
```

```
7275

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

```
% ./testforkwait
```

7275

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

concurrent

7276

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

7

Assume OS gives CPU to parent

```
% ./testforkwait
```

7275

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

7276 (pointer to fork return)

7276

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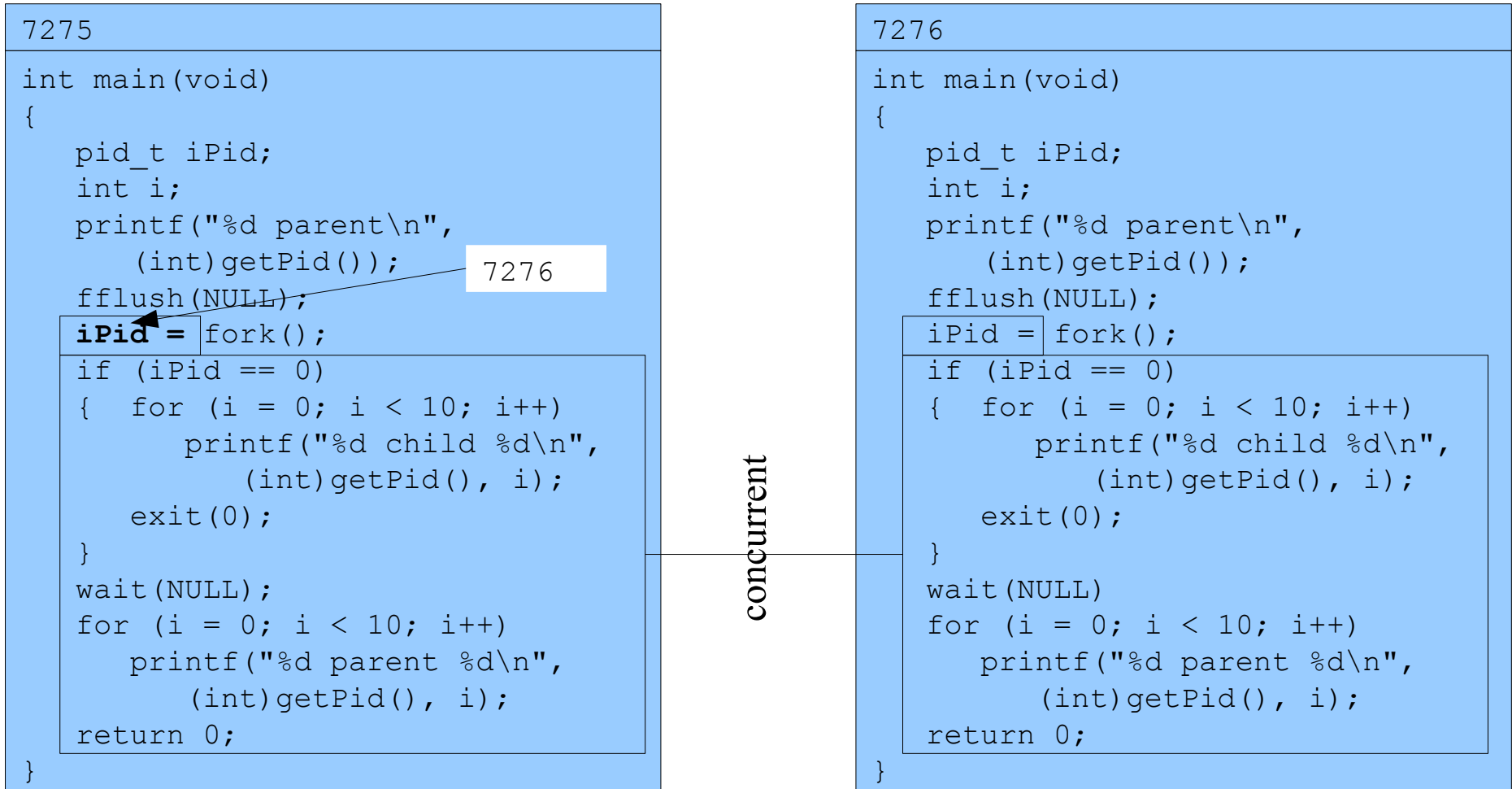
concurrent

8

Assume OS gives CPU to parent

```
% ./testforkwait
```

7275

```
int main(void)
{
   pid_t iPid;
   int i;
   printf("%d parent\n",
      (int)getPid());        7276
   fflush(NULL);
   iPid = fork();
   if (iPid == 0)
   {   for (i = 0; i < 10; i++)
         printf("%d child %d\n",
            (int)getPid(), i);
      exit(0);
   }
   wait(NULL);
   for (i = 0; i < 10; i++)
      printf("%d parent %d\n",
         (int)getPid(), i);
   return 0;
}
```

7276

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

concurrent

Assume OS gives CPU to parent

```
% ./testforkwait
```

7275

```
int main(void)
{
   pid_t iPid;
   int i;
   printf("%d parent\n",
       (int)getPid());
   fflush(NULL);
   iPid = fork();        7276
   if (iPid == 0)
   {   for (i = 0; i < 10; i++)
         printf("%d child %d\n",
            (int)getPid(), i);
       exit(0);
   }
   wait(NULL);
   for (i = 0; i < 10; i++)
      printf("%d parent %d\n",
         (int)getPid(), i);
   return 0;
}
```
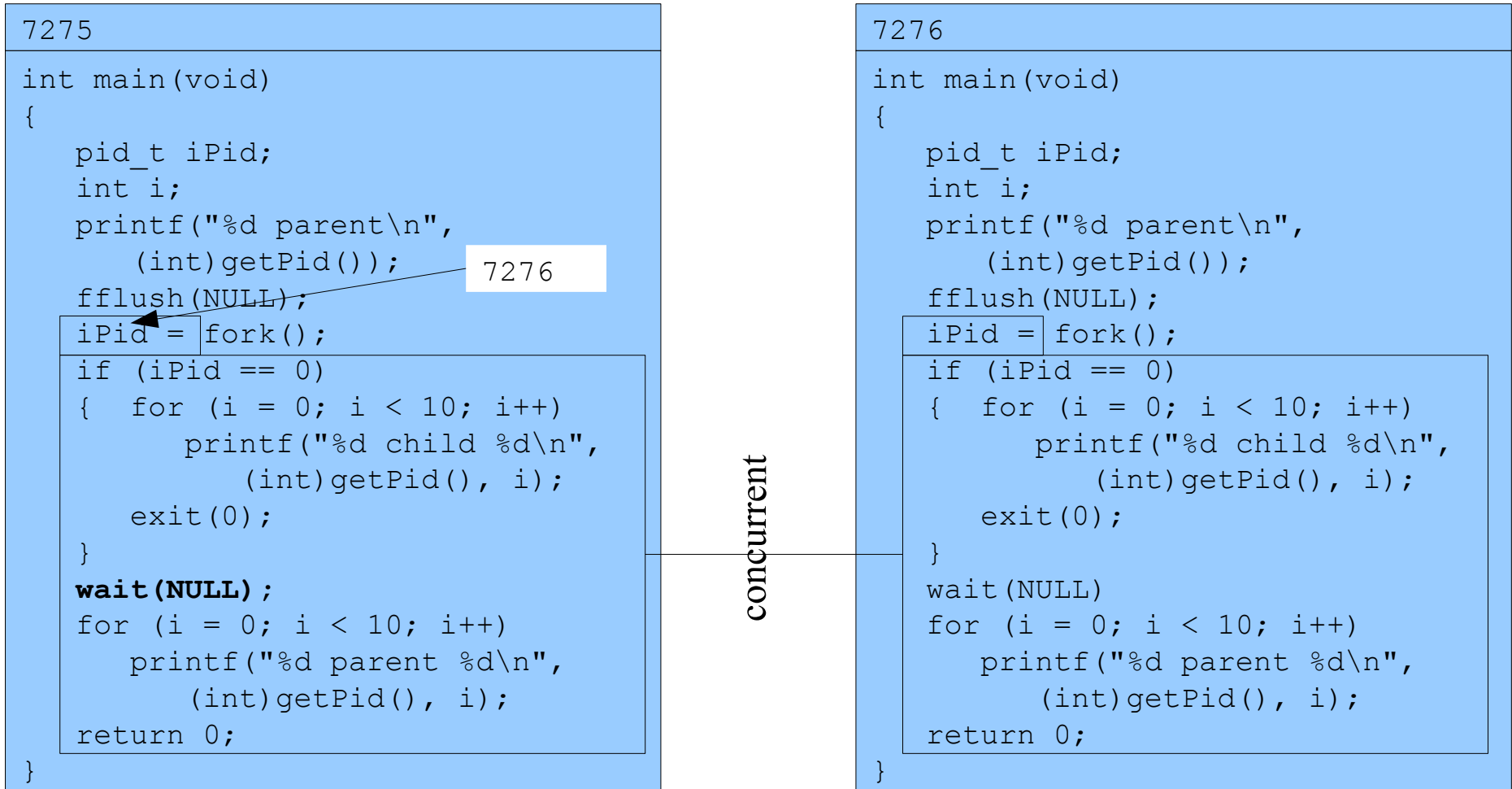
7276

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

concurrent

```
% ./testforkwait
```
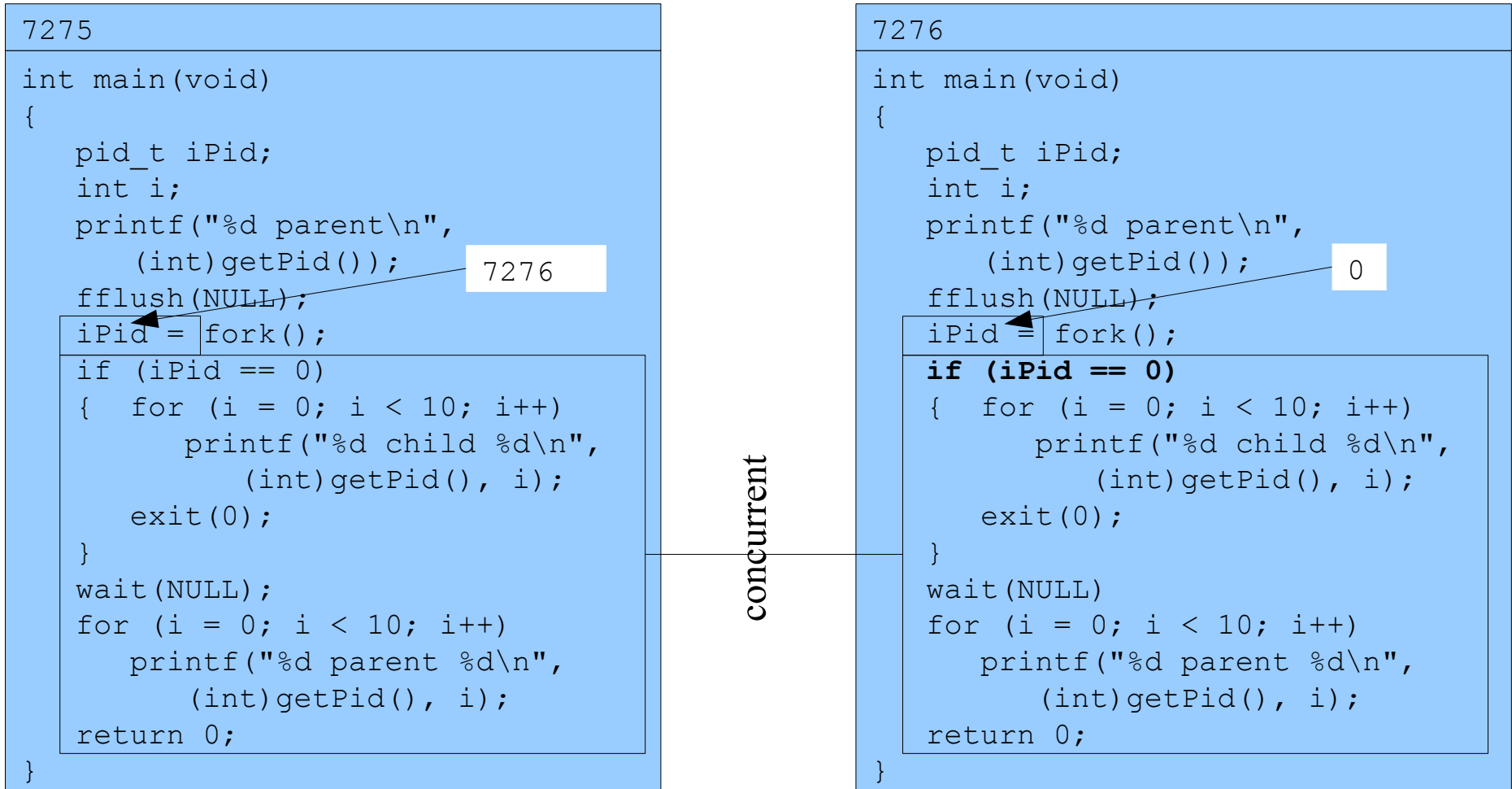
7275

```
int main(void)
{
   pid_t iPid;
   int i;
   printf("%d parent\n",
      (int)getPid());          7276
   fflush(NULL);
   iPid = fork();
   if (iPid == 0)
   {  for (i = 0; i < 10; i++)
         printf("%d child %d\n",
            (int)getPid(), i);
      exit(0);
   }
   wait(NULL);
   for (i = 0; i < 10; i++)
      printf("%d parent %d\n",
         (int)getPid(), i);
   return 0;
}
```

7276

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

concurrent

11

```
% ./testforkwait
```

7275

```
int main(void)
{
   pid_t iPid;
   int i;
   printf("%d parent\n",
      (int)getPid());      [7276]
   fflush(NULL);
   iPid = fork();
   if (iPid == 0)
   {  for (i = 0; i < 10; i++)
         printf("%d child %d\n",
            (int)getPid(), i);
      exit(0);
   }
   wait(NULL);
   for (i = 0; i < 10; i++)
      printf("%d parent %d\n",
         (int)getPid(), i);
   return 0;
}
```

7276

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

concurrent

12

```
% ./testforkwait
```

7275

```
int main(void)
{
   pid_t iPid;
   int i;
   printf("%d parent\n",
      (int)getPid());
   fflush(NULL);
   iPid = fork();          7276
   if (iPid == 0)
   {  for (i = 0; i < 10; i++)
         printf("%d child %d\n",
            (int)getPid(), i);
      exit(0);
   }
   wait(NULL);
   for (i = 0; i < 10; i++)
      printf("%d parent %d\n",
         (int)getPid(), i);
   return 0;
}
```

7276

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

concurrent

Writes:
7276 child 0 ... 9

13

```
% ./testforkwait
```

7275

```
int main(void)
{
   pid_t iPid;
   int i;
   printf("%d parent\n",
       (int)getPid());          7276
   fflush(NULL);
   iPid = fork();
   if (iPid == 0)
   {   for (i = 0; i < 10; i++)
          printf("%d child %d\n",
              (int)getPid(), i);
       exit(0);
   }
   wait(NULL);
   for (i = 0; i < 10; i++)
      printf("%d parent %d\n",
          (int)getPid(), i);
   return 0;
}
```

concurrent

7276

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

14

```
% ./testforkwait
```

```
7275

int main(void)
{
   pid_t iPid;
   int i;
   printf("%d parent\n",
      (int)getPid());          7276
   fflush(NULL);
   iPid = fork();
   if (iPid == 0)
   {  for (i = 0; i < 10; i++)
         printf("%d child %d\n",
            (int)getPid(), i);
      exit(0);
   }
   wait(NULL);
   for (i = 0; i < 10; i++)
      printf("%d parent %d\n",
         (int)getPid(), i);
   return 0;
}
```

Writes:
```
7275 parent 0 ... 9
```

```
% ./testforkwait
```

7275

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

7276

COS 217: Introduction to Programming Systems
Trace of testforkwait

%