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

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

```
% ./testforkloop
```

```
25667

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);
   }
   for (i = 0; i < 10; i++)
      printf("%d parent %d\n",
          (int)getPid(), i);
   return 0;
}
```

```
% ./testforkloop
```

```
25667

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);
   }
   for (i = 0; i < 10; i++)
      printf("%d parent %d\n",
         (int)getPid(), i);
   return 0;
}
```

```
% ./testforkloop
```

```
25667
```
```
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);
   }
   for (i = 0; i < 10; i++)
      printf("%d parent %d\n",
         (int)getPid(), i);
   return 0;
}
```

```
% ./testforkloop
```

```
25667

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);
   }
   for (i = 0; i < 10; i++)
      printf("%d parent %d\n",
         (int)getPid(), i);
   return 0;
}
```

Writes:
```
25667 parent
```

```
% ./testforkloop
```

```
25667

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);
   }
   for (i = 0; i < 10; i++)
      printf("%d parent %d\n",
          (int)getPid(), i);
   return 0;
}
```

```
% ./testforkloop
```

```
25667

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);
   }
   for (i = 0; i < 10; i++)
      printf("%d parent %d\n",
         (int)getPid(), i);
   return 0;
}
```

```
25668

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);
   }
   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
```

25667

```
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);
   }
   for (i = 0; i < 10; i++)
      printf("%d parent %d\n",
         (int)getPid(), i);
   return 0;
}
```

25668

25668

```
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);
   }
   for (i = 0; i < 10; i++)
      printf("%d parent %d\n",
         (int)getPid(), i);
   return 0;
}
```

concurrent

Assume OS gives CPU to parent

```
% ./testforkloop
```

```
25667
int main(void)
{
   pid_t iPid;
   int i;
   printf("%d parent\n",
       (int)getPid());          25668
   fflush(NULL);
   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;
}
```

```
25668
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);
   }
   for (i = 0; i < 10; i++)
      printf("%d parent %d\n",
          (int)getPid(), i);
   return 0;
}
```

concurrent

9

Assume OS gives CPU to parent

```
% ./testforkloop
```

25667

```
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);
   }
   for (i = 0; i < 10; i++)
      printf("%d parent %d\n",
         (int)getPid(), i);
   return 0;
}
```

25668

25668

```
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);
   }
   for (i = 0; i < 10; i++)
      printf("%d parent %d\n",
         (int)getPid(), i);
   return 0;
}
```

concurrent

Writes:
```
25667 parent 0 ... 7
```

10

Assume OS gives CPU to child

```
% ./testforkloop
```

```
25667

int main(void)
{
   pid_t iPid;
   int i;
   printf("%d parent\n",
      (int)getPid());          25668
   fflush(NULL);
   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;
}
```

```
25668

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);
   }
   for (i = 0; i < 10; i++)
      printf("%d parent %d\n",
         (int)getPid(), i);
   return 0;
}
```

concurrent

11

Assume OS gives CPU to child

```
% ./testforkloop
```

```
25667

int main(void)
{
   pid_t iPid;
   int i;
   printf("%d parent\n",
      (int)getPid());          25668
   fflush(NULL);
   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

```
25668

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);
   }
   for (i = 0; i < 10; i++)
      printf("%d parent %d\n",
         (int)getPid(), i);
   return 0;
}
```

12

Assume OS gives CPU to child

```
% ./testforkloop
```

```
25667

int main(void)
{
   pid_t iPid;
   int i;
   printf("%d parent\n",
      (int)getPid());        25668
   fflush(NULL);
   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;
}
```

```
25668

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);
   }
   for (i = 0; i < 10; i++)
      printf("%d parent %d\n",
         (int)getPid(), i);
   return 0;
}
```

concurrent

Writes:
```
25668 child 0
```

13

Assume OS gives CPU to parent

```
% ./testforkloop
```

```
25667

int main(void)
{
   pid_t iPid;
   int i;
   printf("%d parent\n",
      (int)getPid());          25668
   fflush(NULL);
   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;
}
```

```
25668

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);
   }
   for (i = 0; i < 10; i++)
      printf("%d parent %d\n",
         (int)getPid(), i);
   return 0;
}
```

concurrent

Writes:

25667 parent 8

14

Assume OS gives CPU to child

```
% ./testforkloop
```

```
25667

int main(void)
{
   pid_t iPid;
   int i;
   printf("%d parent\n",
       (int)getPid());          25668
   fflush(NULL);
   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;
}
```

```
25668

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);
   }
   for (i = 0; i < 10; i++)
      printf("%d parent %d\n",
         (int)getPid(), i);
   return 0;
}
```

concurrent

Writes:
```
25668 child 1
```

15

Assume OS gives CPU to parent

```
% ./testforkloop
```

```
25667

int main(void)
{
   pid_t iPid;
   int i;
   printf("%d parent\n",
      (int)getPid());          25668
   fflush(NULL);
   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;
}
```

```
25668

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);
   }
   for (i = 0; i < 10; i++)
      printf("%d parent %d\n",
         (int)getPid(), i);
   return 0;
}
```

concurrent

Writes:
```
25669 parent 9
```

16

Assume OS gives CPU to parent

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

```
% ./testforkloop
```

25667

```
int main(void)
{
   pid_t iPid;
   int i;
   printf("%d parent\n",
      (int)getPid());        25668
   fflush(NULL);
   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

25668

```
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);
   }
   for (i = 0; i < 10; i++)
      printf("%d parent %d\n",
         (int)getPid(), i);
   return 0;
}
```

17

```
% ./testforkloop
```

```
25668

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);
   }
   for (i = 0; i < 10; i++)
      printf("%d parent %d\n",
         (int)getPid(), i);
   return 0;
}
```

Writes:
```
25668 child 2 ... 9
```

```
% ./testforkloop
```

```
25668
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);
    }
    for (i = 0; i < 10; i++)
        printf("%d parent %d\n",
            (int)getPid(), i);
    return 0;
}
```

19

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

%