

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

```
% gcc217 testforkexecwait.c -o testforkexecwait
```

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

```
% ./testforkexecwait
```

```
int main(int argc, char *argv[])
{
    pid_t iPid;
    for (;;)
    {
        fflush(NULL);
        iPid = fork();
        if (iPid == 0)
        {
            char *apcArgv[2];
            apcArgv[0] = "date";
            apcArgv[1] = NULL;
            execvp("date", apcArgv);
            perror(argv[0]);
            exit(EXIT_FAILURE);
        }
        wait(NULL);
        sleep(3);
    }
}
```

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

```
% ./testforkexecwait
```

```
int main(int argc, char *argv[])
{
    pid_t iPid;
    for (;;)
    {
        fflush(NULL);
        iPid = fork();
        if (iPid == 0)
        {
            char *apcArgv[2];
            apcArgv[0] = "date";
            apcArgv[1] = NULL;
            execvp("date", apcArgv);
            perror(argv[0]);
            exit(EXIT_FAILURE);
        }
        wait(NULL);
        sleep(3);
    }
}
```

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

```
% ./testforkexecwait
```

```
int main(int argc, char *argv[])
{
    pid_t iPid;
    for (;;)
    {
        fflush(NULL);
        iPid = fork();
        if (iPid == 0)
        {
            char *apcArgv[2];
            apcArgv[0] = "date";
            apcArgv[1] = NULL;
            execvp("date", apcArgv);
            perror(argv[0]);
            exit(EXIT_FAILURE);
        }
        wait(NULL);
        sleep(3);
    }
}
```

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

```
% ./testforkexecwait
```

```
int main(int argc, char *argv[])
{
    pid_t iPid;
    for (;;)
    { fflush(NULL) ;
      iPid = fork();
      if (iPid == 0)
      { char *apcArgv[2];
        apcArgv[0] = "date";
        apcArgv[1] = NULL;
        execvp("date", apcArgv);
        perror(argv[0]);
        exit(EXIT_FAILURE);
      }
      wait(NULL);
      sleep(3);
    }
}
```

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

```
% ./testforkexecwait
```

```
int main(int argc, char *argv[])
{
    pid_t iPid;
    for (;;)
    { fflush(NULL);
      iPid = fork();
      if (iPid == 0)
      { char *apcArgv[2];
        apcArgv[0] = "date";
        apcArgv[1] = NULL;
        execvp("date", apcArgv);
        perror(argv[0]);
        exit(EXIT_FAILURE);
      }
      wait(NULL);
      sleep(3);
    }
}
```

concurrent

```
int main(int argc, char *argv[])
{
    pid_t iPid;
    for (;;)
    { fflush(NULL);
      iPid = fork();
      if (iPid == 0)
      { char *apcArgv[2];
        apcArgv[0] = "date";
        apcArgv[1] = NULL;
        execvp("date", apcArgv);
        perror(argv[0]);
        exit(EXIT_FAILURE);
      }
      wait(NULL);
      sleep(3);
    }
}
```

Assume OS gives CPU to parent

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

```
% ./testforkexecwait
```

```
int main(int argc, char *argv[])
{
    pid_t iPid;
    for (;;)
    { fflush(NULL);
      iPid = fork();
      if (iPid == 0)
      { char *apcArgv[2];
        apcArgv[0] = "date";
        apcArgv[1] = NULL;
        execvp("date", apcArgv);
        perror(argv[0]);
        exit(EXIT_FAILURE);
      }
      wait(NULL);
      sleep(3);
    }
}
```

not 0

concurrent

```
int main(int argc, char *argv[])
{
    pid_t iPid;
    for (;;)
    { fflush(NULL);
      iPid = fork();
      if (iPid == 0)
      { char *apcArgv[2];
        apcArgv[0] = "date";
        apcArgv[1] = NULL;
        execvp("date", apcArgv);
        perror(argv[0]);
        exit(EXIT_FAILURE);
      }
      wait(NULL);
      sleep(3);
    }
}
```

Assume OS gives CPU to parent

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

```
% ./testforkexecwait
```

```
int main(int argc, char *argv[])
{
    pid_t iPid;
    for (;;)
    { fflush(NULL);
      iPid = fork();
      if (iPid == 0)
      { char *apcArgv[2];
        apcArgv[0] = "date";
        apcArgv[1] = NULL;
        execvp("date", apcArgv);
        perror(argv[0]);
        exit(EXIT_FAILURE);
      }
      wait(NULL);
      sleep(3);
    }
}
```

not 0

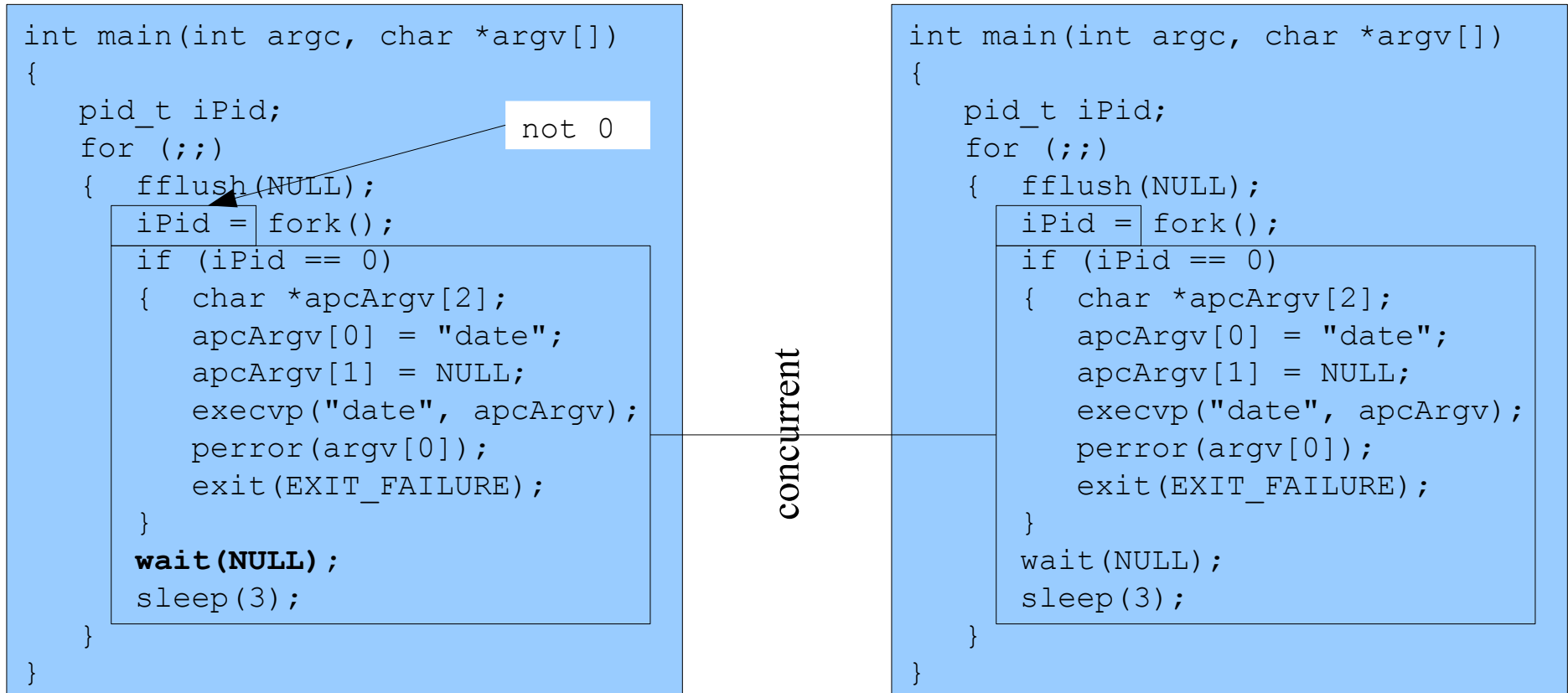
```
int main(int argc, char *argv[])
{
    pid_t iPid;
    for (;;)
    { fflush(NULL);
      iPid = fork();
      if (iPid == 0)
      { char *apcArgv[2];
        apcArgv[0] = "date";
        apcArgv[1] = NULL;
        execvp("date", apcArgv);
        perror(argv[0]);
        exit(EXIT_FAILURE);
      }
      wait(NULL);
      sleep(3);
    }
}
```

concurrent

Assume OS gives CPU to parent

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

```
% ./testforkexecwait
```



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

```
% ./testforkexecwait
```

```
int main(int argc, char *argv[])
{
    pid_t iPid;
    for (;;)
    { fflush(NULL);
      iPid = fork();
      if (iPid == 0)
      { char *apcArgv[2];
        apcArgv[0] = "date";
        apcArgv[1] = NULL;
        execvp("date", apcArgv);
        perror(argv[0]);
        exit(EXIT_FAILURE);
      }
      wait(NULL);
      sleep(3);
    }
}
```

not 0

```
int main(int argc, char *argv[])
{
    pid_t iPid;
    for (;;)
    { fflush(NULL);
      iPid = fork();
      if (iPid == 0)
      { char *apcArgv[2];
        apcArgv[0] = "date";
        apcArgv[1] = NULL;
        execvp("date", apcArgv);
        perror(argv[0]);
        exit(EXIT_FAILURE);
      }
      wait(NULL);
      sleep(3);
    }
}
```

0

concurrent

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

```
% ./testforkexecwait
```

```
int main(int argc, char *argv[])
{
    pid_t iPid;
    for (;;)
    { fflush(NULL);
      iPid = fork();
      if (iPid == 0)
      { char *apcArgv[2];
        apcArgv[0] = "date";
        apcArgv[1] = NULL;
        execvp("date", apcArgv);
        perror(argv[0]);
        exit(EXIT_FAILURE);
      }
      wait(NULL);
      sleep(3);
    }
}
```

not 0

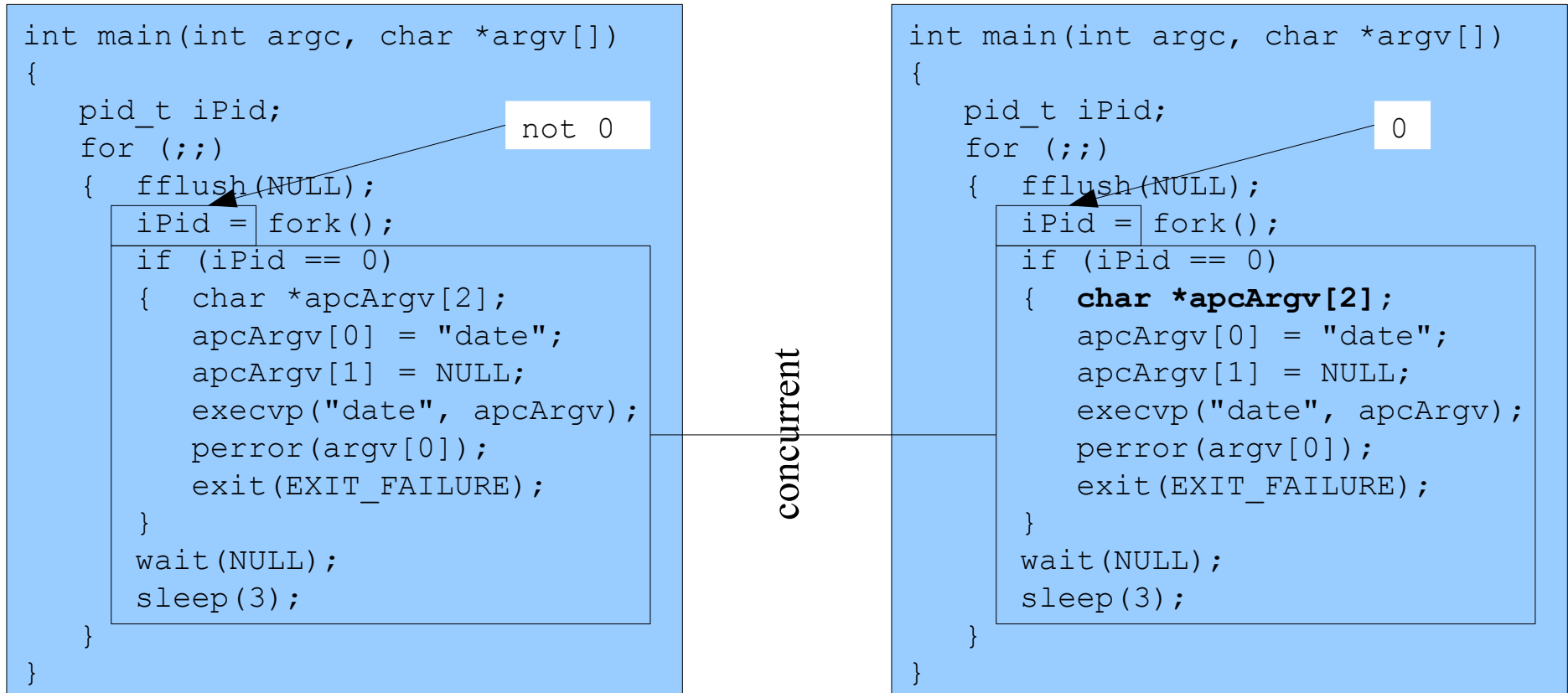
```
int main(int argc, char *argv[])
{
    pid_t iPid;
    for (;;)
    { fflush(NULL);
      iPid = fork();
      if (iPid == 0)
      { char *apcArgv[2];
        apcArgv[0] = "date";
        apcArgv[1] = NULL;
        execvp("date", apcArgv);
        perror(argv[0]);
        exit(EXIT_FAILURE);
      }
      wait(NULL);
      sleep(3);
    }
}
```

0

concurrent

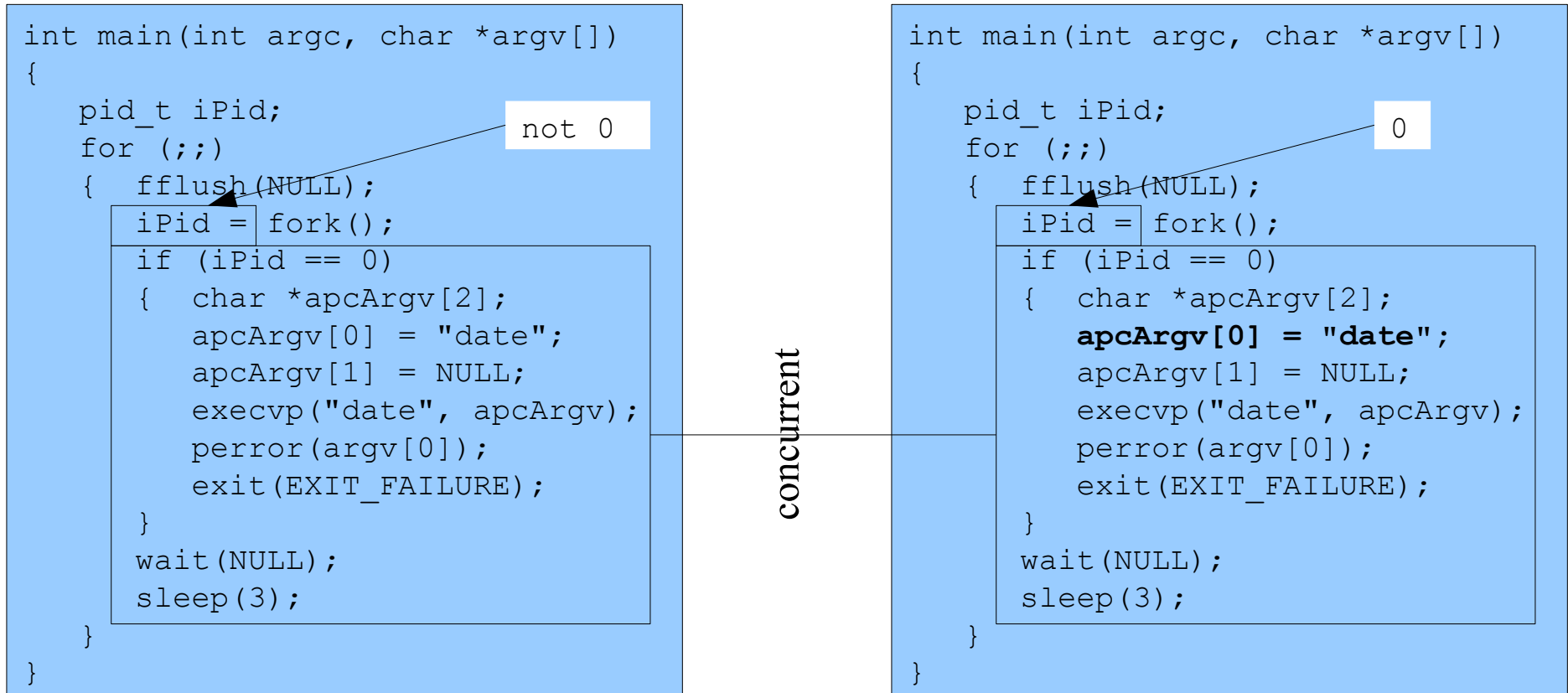
Princeton University
COS 217: Introduction to Programming Systems
Trace of testforkexecwait

```
% ./testforkexecwait
```



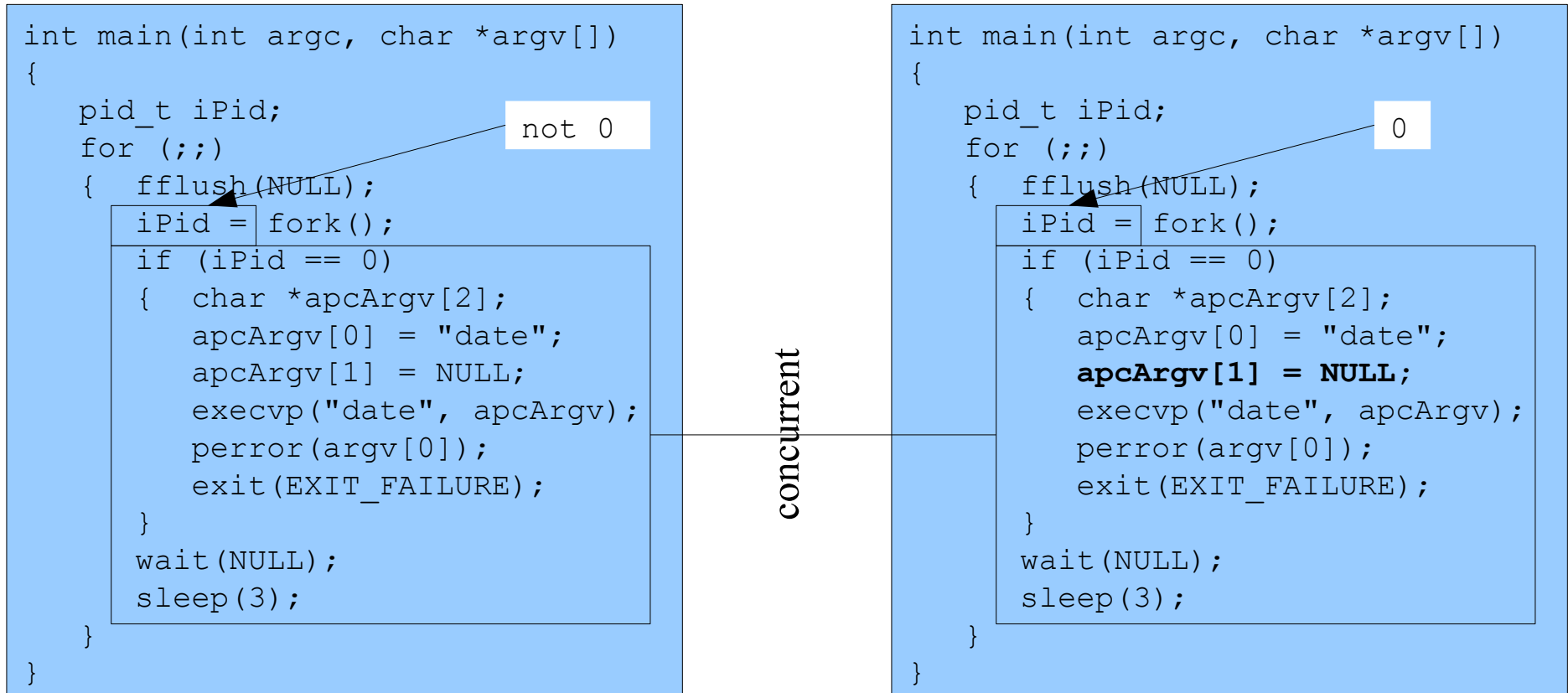
Princeton University
COS 217: Introduction to Programming Systems
Trace of testforkexecwait

```
% ./testforkexecwait
```



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

```
% ./testforkexecwait
```



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

```
% ./testforkexecwait
```

```
int main(int argc, char *argv[])
{
    pid_t iPid;
    for (;;)
    { fflush(NULL);
      iPid = fork();
      if (iPid == 0)
      { char *apcArgv[2];
        apcArgv[0] = "date";
        apcArgv[1] = NULL;
        execvp("date", apcArgv);
        perror(argv[0]);
        exit(EXIT_FAILURE);
      }
      wait(NULL);
      sleep(3);
    }
}
```

not 0

```
int main(int argc, char *argv[])
{
    pid_t iPid;
    for (;;)
    { fflush(NULL);
      iPid = fork();
      if (iPid == 0)
      { char *apcArgv[2];
        apcArgv[0] = "date";
        apcArgv[1] = NULL;
        execvp("date", apcArgv);
        perror(argv[0]);
        exit(EXIT_FAILURE);
      }
      wait(NULL);
      sleep(3);
    }
}
```

0

concurrent

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

```
% ./testforkexecwait
```

```
int main(int argc, char *argv[])
{
    pid_t iPid;
    for (;;)
    {
        fflush(NULL);
        iPid = fork();
        if (iPid == 0)
        {
            char *apcArgv[2];
            apcArgv[0] = "date";
            apcArgv[1] = NULL;
            execvp("date", apcArgv);
            perror(argv[0]);
            exit(EXIT_FAILURE);
        }
        wait(NULL);
        sleep(3);
    }
}
```

not 0

```
int main(int argc, char *argv[])
{
    Date
    program

    return 0;
}
```

concurrent

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

```
% ./testforkexecwait
```

```
int main(int argc, char *argv[])
{
    pid_t iPid;
    for (;;)
    {   fflush(NULL);
        iPid = fork();
        if (iPid == 0)
        {   char *apcArgv[2];
            apcArgv[0] = "date";
            apcArgv[1] = NULL;
            execvp("date", apcArgv);
            perror(argv[0]);
            exit(EXIT_FAILURE);
        }
        wait(NULL);
        sleep(3);
    }
}
```

not 0

```
int main(int argc, char *argv[])
{
    Date  
program

    return 0;
}
```

concurrent

Writes the current date/time

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

```
% ./testforkexecwait
```

```
int main(int argc, char *argv[])
{
    pid_t iPid;
    for (;;)
    {
        fflush(NULL);
        iPid = fork();
        if (iPid == 0)
        {
            char *apcArgv[2];
            apcArgv[0] = "date";
            apcArgv[1] = NULL;
            execvp("date", apcArgv);
            perror(argv[0]);
            exit(EXIT_FAILURE);
        }
        wait(NULL);
        sleep(3);
    }
}
```

not 0

concurrent

```
int main(int argc, char *argv[])
{
    Date
    program
    return 0;
}
```

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

```
% ./testforkexecwait
```

```
int main(int argc, char *argv[])
{
    pid_t iPid;
    for (;;)
    {
        fflush(NULL);
        iPid = fork();
        if (iPid == 0)
        {
            char *apcArgv[2];
            apcArgv[0] = "date";
            apcArgv[1] = NULL;
            execvp("date", apcArgv);
            perror(argv[0]);
            exit(EXIT_FAILURE);
        }
        wait(NULL);
        sleep(3);
    }
}
```

not 0

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

```
% ./testforkexecwait
```

```
int main(int argc, char *argv[])
{
    pid_t iPid;
    for (;;)
    {
        fflush(NULL);
        iPid = fork();
        if (iPid == 0)
        {
            char *apcArgv[2];
            apcArgv[0] = "date";
            apcArgv[1] = NULL;
            execvp("date", apcArgv);
            perror(argv[0]);
            exit(EXIT_FAILURE);
        }
        wait(NULL);
        sleep(3);
    }
}
```

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

```
% ./testforkexecwait
```

```
int main(int argc, char *argv[])
{
    pid_t iPid;
    for (;;)
    { fflush(NULL) ;
      iPid = fork();
      if (iPid == 0)
      { char *apcArgv[2];
        apcArgv[0] = "date";
        apcArgv[1] = NULL;
        execvp("date", apcArgv);
        perror(argv[0]);
        exit(EXIT_FAILURE);
      }
      wait(NULL);
      sleep(3);
    }
}
```

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

```
% ./testforkexecwait
```

```
int main(int argc, char *argv[])
{
    pid_t iPid;
    for (;;)
    { fflush(NULL);
      iPid = fork();
      if (iPid == 0)
      { char *apcArgv[2];
        apcArgv[0] = "date";
        apcArgv[1] = NULL;
        execvp("date", apcArgv);
        perror(argv[0]);
        exit(EXIT_FAILURE);
      }
      wait(NULL);
      sleep(3);
    }
}
```

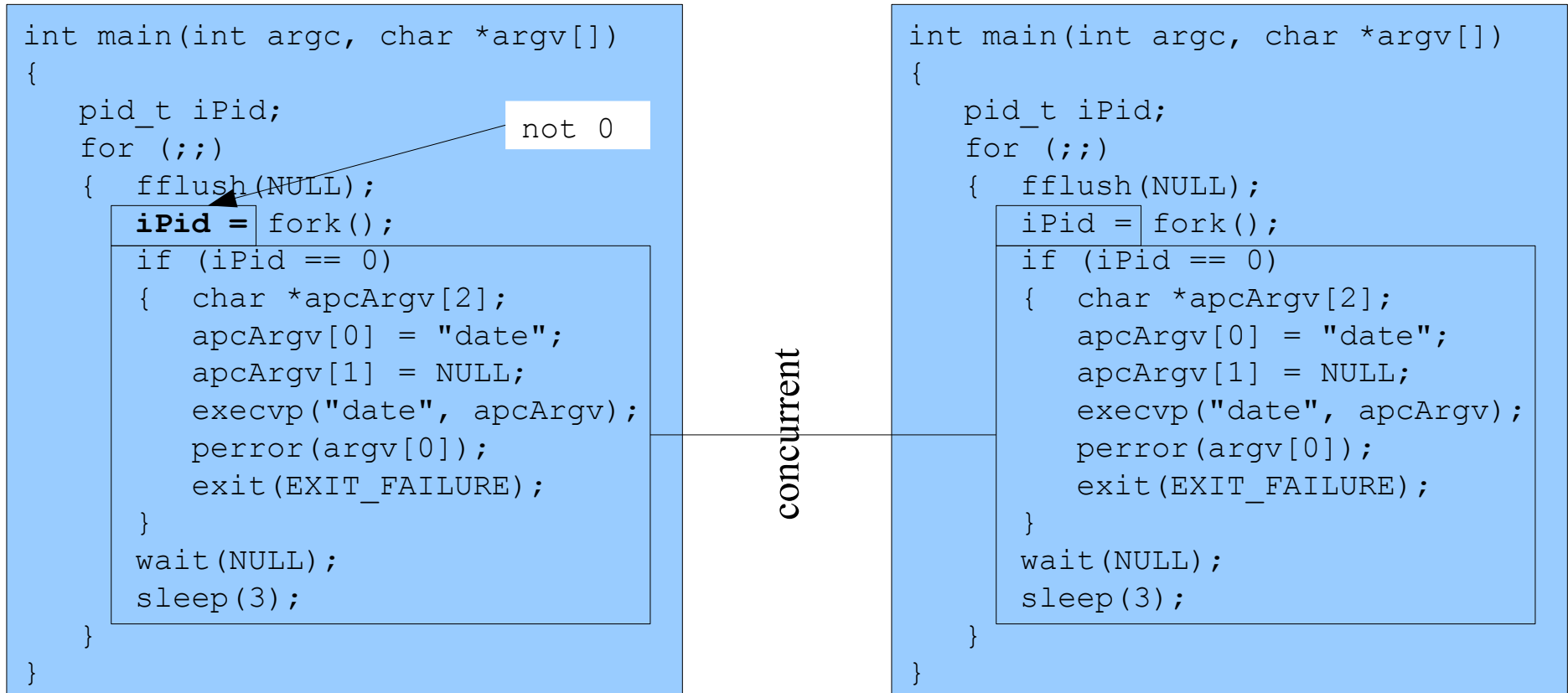
concurrent

```
int main(int argc, char *argv[])
{
    pid_t iPid;
    for (;;)
    { fflush(NULL);
      iPid = fork();
      if (iPid == 0)
      { char *apcArgv[2];
        apcArgv[0] = "date";
        apcArgv[1] = NULL;
        execvp("date", apcArgv);
        perror(argv[0]);
        exit(EXIT_FAILURE);
      }
      wait(NULL);
      sleep(3);
    }
}
```

Assume OS gives CPU to parent

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

```
% ./testforkexecwait
```



Assume OS gives CPU to parent

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

```
% ./testforkexecwait
```

```
int main(int argc, char *argv[])
{
    pid_t iPid;
    for (;;)
    { fflush(NULL);
      iPid = fork();
      if (iPid == 0)
      { char *apcArgv[2];
        apcArgv[0] = "date";
        apcArgv[1] = NULL;
        execvp("date", apcArgv);
        perror(argv[0]);
        exit(EXIT_FAILURE);
      }
      wait(NULL);
      sleep(3);
    }
}
```

not 0

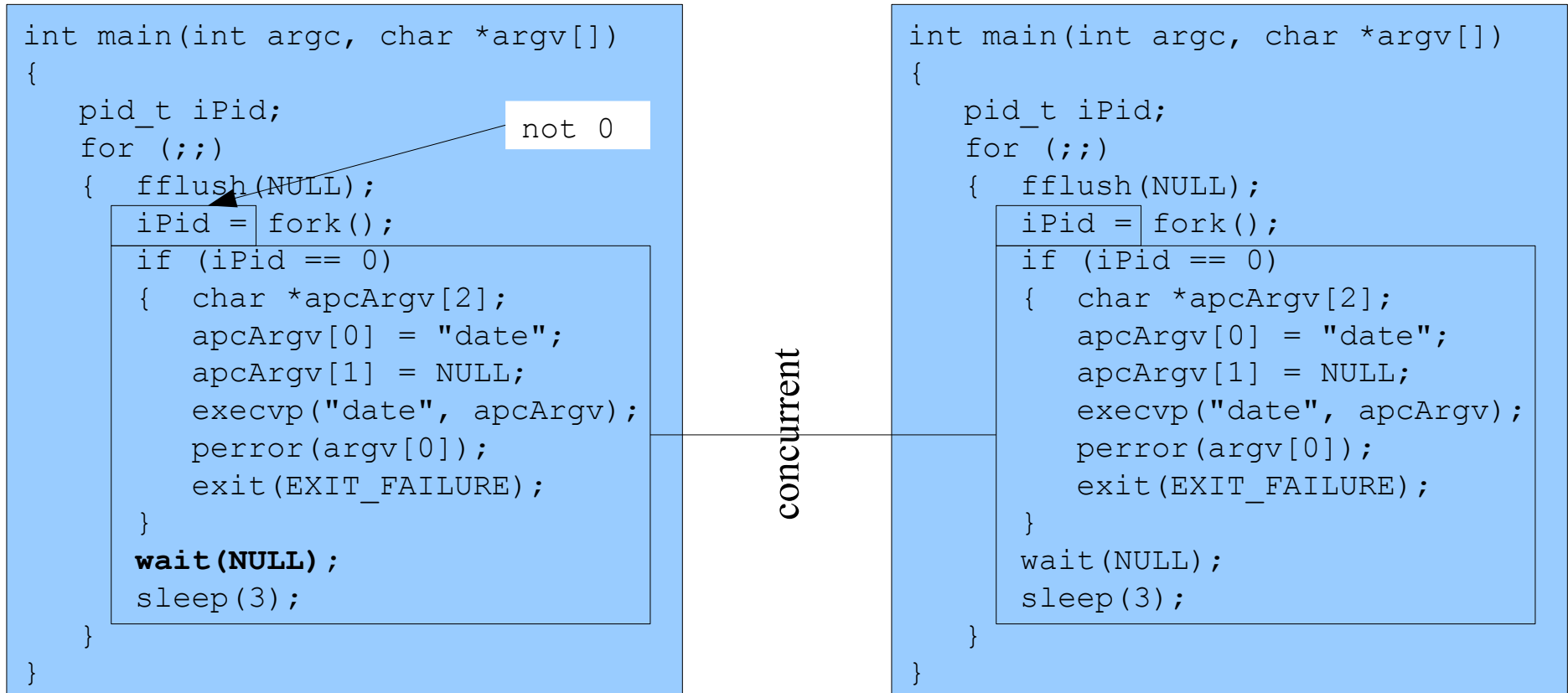
concurrent

```
int main(int argc, char *argv[])
{
    pid_t iPid;
    for (;;)
    { fflush(NULL);
      iPid = fork();
      if (iPid == 0)
      { char *apcArgv[2];
        apcArgv[0] = "date";
        apcArgv[1] = NULL;
        execvp("date", apcArgv);
        perror(argv[0]);
        exit(EXIT_FAILURE);
      }
      wait(NULL);
      sleep(3);
    }
}
```

Assume OS gives CPU to parent

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

```
% ./testforkexecwait
```



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

```
% ./testforkexecwait
```

```
int main(int argc, char *argv[])
{
    pid_t iPid;
    for (;;)
    {
        fflush(NULL);
        iPid = fork();
        if (iPid == 0)
        {
            char *apcArgv[2];
            apcArgv[0] = "date";
            apcArgv[1] = NULL;
            execvp("date", apcArgv);
            perror(argv[0]);
            exit(EXIT_FAILURE);
        }
        wait(NULL);
        sleep(3);
    }
}
```

not 0

```
int main(int argc, char *argv[])
{
    pid_t iPid;
    for (;;)
    {
        fflush(NULL);
        iPid = fork();
        if (iPid == 0)
        {
            char *apcArgv[2];
            apcArgv[0] = "date";
            apcArgv[1] = NULL;
            execvp("date", apcArgv);
            perror(argv[0]);
            exit(EXIT_FAILURE);
        }
        wait(NULL);
        sleep(3);
    }
}
```

0

concurrent

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

```
% ./testforkexecwait
```

```
int main(int argc, char *argv[])
{
    pid_t iPid;
    for (;;)
    { fflush(NULL);
      iPid = fork();
      if (iPid == 0)
      { char *apcArgv[2];
        apcArgv[0] = "date";
        apcArgv[1] = NULL;
        execvp("date", apcArgv);
        perror(argv[0]);
        exit(EXIT_FAILURE);
      }
      wait(NULL);
      sleep(3);
    }
}
```

not 0

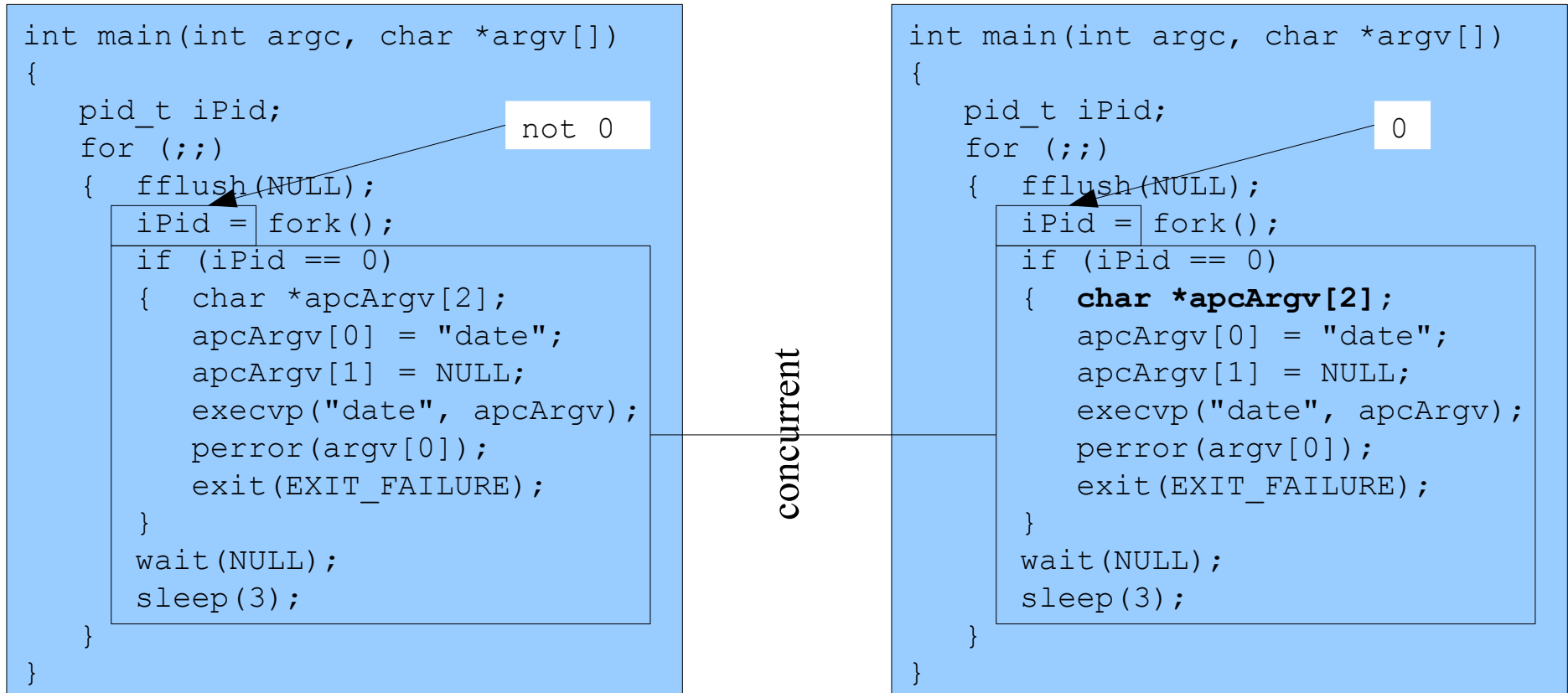
```
int main(int argc, char *argv[])
{
    pid_t iPid;
    for (;;)
    { fflush(NULL);
      iPid = fork();
      if (iPid == 0)
      { char *apcArgv[2];
        apcArgv[0] = "date";
        apcArgv[1] = NULL;
        execvp("date", apcArgv);
        perror(argv[0]);
        exit(EXIT_FAILURE);
      }
      wait(NULL);
      sleep(3);
    }
}
```

0

concurrent

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

```
% ./testforkexecwait
```



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

```
% ./testforkexecwait
```

```
int main(int argc, char *argv[])
{
    pid_t iPid;
    for (;;)
    {
        fflush(NULL);
        iPid = fork();
        if (iPid == 0)
        {
            char *apcArgv[2];
            apcArgv[0] = "date";
            apcArgv[1] = NULL;
            execvp("date", apcArgv);
            perror(argv[0]);
            exit(EXIT_FAILURE);
        }
        wait(NULL);
        sleep(3);
    }
}
```

not 0

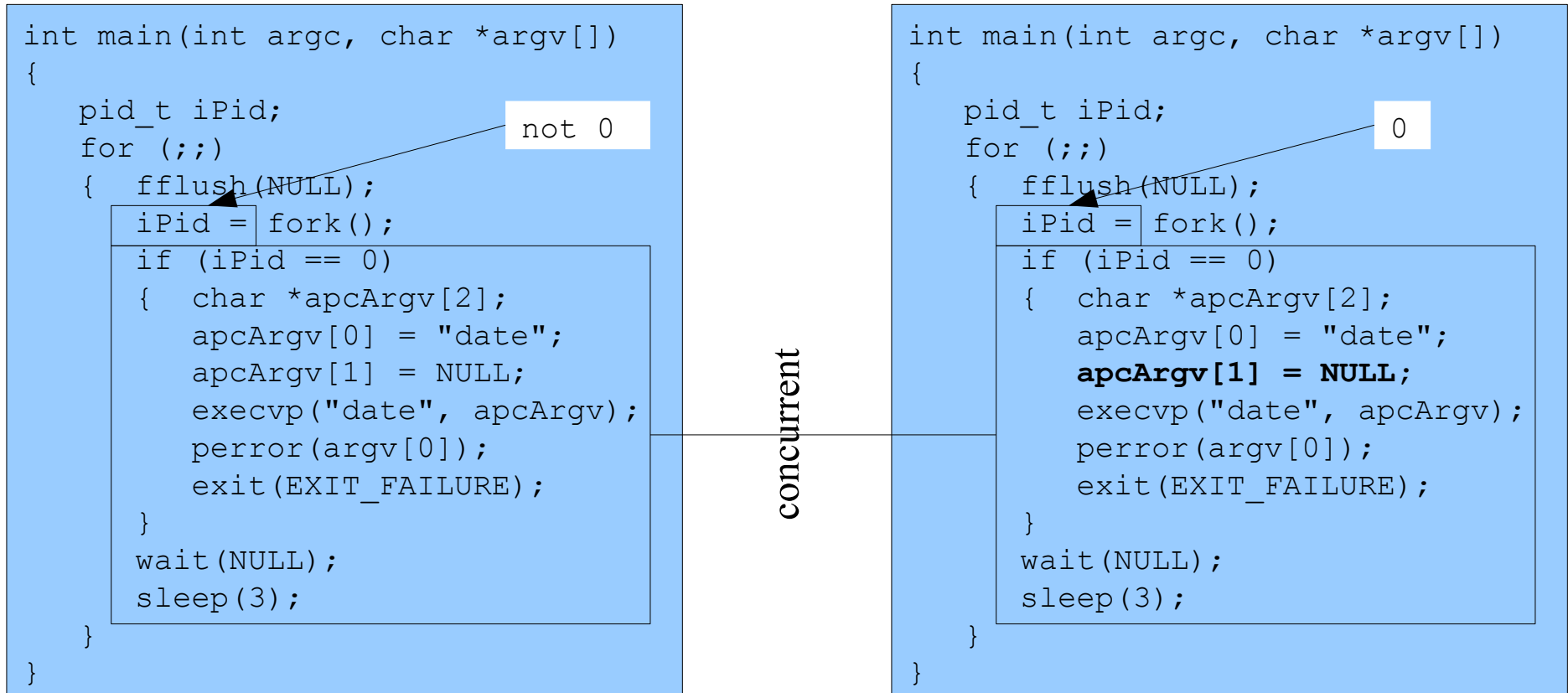
```
int main(int argc, char *argv[])
{
    pid_t iPid;
    for (;;)
    {
        fflush(NULL);
        iPid = fork();
        if (iPid == 0)
        {
            char *apcArgv[2];
            apcArgv[0] = "date";
            apcArgv[1] = NULL;
            execvp("date", apcArgv);
            perror(argv[0]);
            exit(EXIT_FAILURE);
        }
        wait(NULL);
        sleep(3);
    }
}
```

0

concurrent

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

```
% ./testforkexecwait
```



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

```
% ./testforkexecwait
```

```
int main(int argc, char *argv[])
{
    pid_t iPid;
    for (;;)
    { fflush(NULL);
      iPid = fork();
      if (iPid == 0)
      { char *apcArgv[2];
        apcArgv[0] = "date";
        apcArgv[1] = NULL;
        execvp("date", apcArgv);
        perror(argv[0]);
        exit(EXIT_FAILURE);
      }
      wait(NULL);
      sleep(3);
    }
}
```

not 0

```
int main(int argc, char *argv[])
{
    pid_t iPid;
    for (;;)
    { fflush(NULL);
      iPid = fork();
      if (iPid == 0)
      { char *apcArgv[2];
        apcArgv[0] = "date";
        apcArgv[1] = NULL;
        execvp("date", apcArgv);
        perror(argv[0]);
        exit(EXIT_FAILURE);
      }
      wait(NULL);
      sleep(3);
    }
}
```

0

concurrent

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

```
% ./testforkexecwait
```

```
int main(int argc, char *argv[])
{
    pid_t iPid;
    for (;;)
    { fflush(NULL);
      iPid = fork();
      if (iPid == 0)
      { char *apcArgv[2];
        apcArgv[0] = "date";
        apcArgv[1] = NULL;
        execvp("date", apcArgv);
        perror(argv[0]);
        exit(EXIT_FAILURE);
      }
      wait(NULL);
      sleep(3);
    }
}
```

not 0

```
iPid = fork();
if (iPid == 0)
{ char *apcArgv[2];
  apcArgv[0] = "date";
  apcArgv[1] = NULL;
  execvp("date", apcArgv);
  perror(argv[0]);
  exit(EXIT_FAILURE);
}
wait(NULL);
sleep(3);
```

```
int main(int argc, char *argv[])
{
    Date
    program

    return 0;
}
```

concurrent

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

```
% ./testforkexecwait
```

```
int main(int argc, char *argv[])
{
    pid_t iPid;
    for (;;)
    {
        fflush(NULL);
        iPid = fork();
        if (iPid == 0)
        {
            char *apcArgv[2];
            apcArgv[0] = "date";
            apcArgv[1] = NULL;
            execvp("date", apcArgv);
            perror(argv[0]);
            exit(EXIT_FAILURE);
        }
        wait(NULL);
        sleep(3);
    }
}
```

not 0

```
if (iPid == 0)
{
    char *apcArgv[2];
    apcArgv[0] = "date";
    apcArgv[1] = NULL;
    execvp("date", apcArgv);
    perror(argv[0]);
    exit(EXIT_FAILURE);
}
```

```
int main(int argc, char *argv[])
{
    Date program
    return 0;
}
```

concurrent

Writes the current date/time

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

```
% ./testforkexecwait
```

```
int main(int argc, char *argv[])
{
    pid_t iPid;
    for (;;)
    { fflush(NULL);
      iPid = fork();
      if (iPid == 0)
      { char *apcArgv[2];
        apcArgv[0] = "date";
        apcArgv[1] = NULL;
        execvp("date", apcArgv);
        perror(argv[0]);
        exit(EXIT_FAILURE);
      }
      wait(NULL);
      sleep(3);
    }
}
```

not 0

concurrent

```
int main(int argc, char *argv[])
{
    Date
    program
    return 0;
}
```

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

```
% ./testforkexecwait
```

```
int main(int argc, char *argv[])
{
    pid_t iPid;
    for (;;)
    {
        fflush(NULL);
        iPid = fork();
        if (iPid == 0)
        {
            char *apcArgv[2];
            apcArgv[0] = "date";
            apcArgv[1] = NULL;
            execvp("date", apcArgv);
            perror(argv[0]);
            exit(EXIT_FAILURE);
        }
        wait(NULL);
        sleep(3);
    }
}
```

not 0

Copyright © 2016 by Robert M. Dondero, Jr.

35

Repeats infinitely