

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(stdin);
        fflush(stdout);
        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(stdin);
        fflush(stdout);
        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(stdin);
        fflush(stdout);
        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(stdin);
        fflush(stdout);
        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(stdin);
        fflush(stdout);
        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(stdin);
        fflush(stdout);
        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(stdin);
        fflush(stdout);
        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(stdin);
        fflush(stdout);
        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(stdin);
        fflush(stdout);
        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
```

```
int main(int argc, char *argv[])
{
    pid_t iPid;
    for (;;)
    {
        fflush(stdin);
        fflush(stdout);
        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(stdin);
        fflush(stdout);
        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(stdin);
        fflush(stdout);
        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(stdin);
        fflush(stdout);
        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(stdin);
        fflush(stdout);
        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(stdin);
        fflush(stdout);
        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(stdin);
        fflush(stdout);
        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(stdin);
        fflush(stdout);
        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(stdin);
        fflush(stdout);
        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(stdin);
        fflush(stdout);
        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(stdin);
        fflush(stdout);
        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(stdin);
        fflush(stdout);
        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(stdin);
        fflush(stdout);
        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(stdin);
        fflush(stdout);
        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

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

```
% ./testforkexecwait
```

```
int main(int argc, char *argv[])
{
    pid_t iPid;
    for (;;)
    {
        fflush(stdin);
        fflush(stdout);
        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(stdin);
        fflush(stdout);
        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

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

```
% ./testforkexecwait
```

```
int main(int argc, char *argv[])
{
    pid_t iPid;
    for (;;)
    {
        fflush(stdin);
        fflush(stdout);
        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)

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(stdin);
        fflush(stdout);
        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)

concurrent

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

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(stdin);
        fflush(stdout);
        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)

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(stdin);
        fflush(stdout);
        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(stdin);
        fflush(stdout);
        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(stdin);
        fflush(stdout);
        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(stdin);
        fflush(stdout);
        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(stdin);
        fflush(stdout);
        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(stdin);
        fflush(stdout);
        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(stdin);
        fflush(stdout);
        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(stdin);
        fflush(stdout);
        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(stdin);
        fflush(stdout);
        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(stdin);
        fflush(stdout);
        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(stdin);
        fflush(stdout);
        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(stdin);
        fflush(stdout);
        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(stdin);
        fflush(stdout);
        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); }

wait(NULL);
sleep(3);

concurrent

```
int main(int argc, char *argv[])
{
    pid_t iPid;
    for (;;)
    {
        fflush(stdin);
        fflush(stdout);
        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

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(stdin);
        fflush(stdout);
        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);
}
```

concurrent

```
int main(int argc, char *argv[])
{
    pid_t iPid;
    for (;;)
    {
        fflush(stdin);
        fflush(stdout);
        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

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

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

```
% ./testforkexecwait
```

```
int main(int argc, char *argv[])
{
    pid_t iPid;
    for (;;)
    {
        fflush(stdin);
        fflush(stdout);
        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(stdin);
        fflush(stdout);
        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

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

```
% ./testforkexecwait
```

```
int main(int argc, char *argv[])
{
    pid_t iPid;
    for (;;)
    {
        fflush(stdin);
        fflush(stdout);
        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(stdin);
        fflush(stdout);
        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

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

```
% ./testforkexecwait
```

```
int main(int argc, char *argv[])
{
    pid_t iPid;
    for (;;)
    {
        fflush(stdin);
        fflush(stdout);
        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(stdin);
        fflush(stdout);
        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(stdin);
        fflush(stdout);
        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(stdin);
        fflush(stdout);
        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

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

```
% ./testforkexecwait
```

```
int main(int argc, char *argv[])
{
    pid_t iPid;
    for (;;)
    {
        fflush(stdin);
        fflush(stdout);
        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)

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(stdin);
        fflush(stdout);
        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)

concurrent

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

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(stdin);
        fflush(stdout);
        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)

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(stdin);
        fflush(stdout);
        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 © 2019 by Robert M. Dondero, Jr.

37

Repeats infinitely