

```
#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>
```

```
int main()
{
    pid_t pid, pidd;
```

```
    /* fork a child process */
    pid = fork();
```

```
    if (pid < 0) { /* error occurred */
        fprintf(stderr, "Fork Failed");
        return 1;
    }
```

```
    else if (pid == 0) { /* child process */
        pid = getpid();
        printf("child: pid = %d", pid); /* A */
        printf("child: pid = %d", pid); /* B */
    }
```

```
    else { /* parent process */
        pid = getpid();
```

```
        printf("parent: pid = %d", pid); /* C */
        printf("parent: pid = %d", pid); /* D */
        wait(NULL);
    }
```

```
    return 0;
}
```

Figure 3.29 What are the pid values?

3.14 What are the benefits and the disadvantages of each of the following? Consider both the system level and the programmer level.

- Synchronous and asynchronous communication
- Automatic and explicit buffering
- Send by copy and send by reference
- Fixed-sized and variable-sized messages

Programming Problems

3.15 The Fibonacci sequence is the series of numbers 0, 1, 1, 2, 3, 5, 8, Formally, it can be expressed as:

$$\begin{aligned} fib_0 &= 0 \\ fib_1 &= 1 \\ fib_n &= fib_{n-1} + fib_{n-2} \end{aligned}$$