

به نام خدا

علی شیخ عطار ۹۹۵۴۲۲۲۲

```
int main() {  
    int pipefd[2];  
    char message[] = "Hello from parent!";  
    char buffer[100];  
  
    if (pipe(pipefd) == -1) {  
        perror("pipe");  
        return 1;  
    }  
}
```

در ابتدا کد بالا اجرا میشود

- pipefd آرایه ی برای ذخیره ی file descriptor هایی است که یکی برای write و یکی برای read می باشد.
- Message همان پیامی است که قرار است والد بنویسد.
- Buffer آرایه است که فرزند پیامی که می خواند را در آن ذخیره میکند.

```
pid_t pid = fork();  
  
if (pid == -1) {  
    perror("fork");  
    return 1;  
} else if (pid == 0) {
```

در اینجا fork صورت می گیرد و فرزند ساخته می شود و در صورت خطا ارور مربوطه لاگ میشود

```

} else if (pid == 0) {

    close(pipefd[1]);

    ssize_t bytesRead = read(pipefd[0], buffer, sizeof(buffer));
    if (bytesRead == -1) {
        perror("read");
        return 1;
    }

    buffer[bytesRead] = '\0';
    printf("Child received: %s\n", buffer);

    close(pipefd[0]);
}

```

کد بالا با توجه به شرط، فرزند اجرا میکند و ابتدا پایپ مربوط به نوشتن را میبندد و نشان میدهد قصد خواندن را دارد. در خط بعدی از انتهای خواندن پایپ، فایل را میخواند و در `buffer` ذخیره میکند و در صورتی که دیتایی وجود نداشت، `wait` میکند تا `write` صورت بگیرد. در صورتی که خروجی `read` که تعداد بایت های خوانده شده می باشد منفی یک باشد یعنی عملیات خواندن موفق نبوده. در آخر بایت آخر را صفر می گذارد، آن را چاپ میکند و نتیجه را چاپ میکند.

```

} else {

    close(pipefd[0]);

    ssize_t bytesWritten = write(pipefd[1], message, strlen(message));
    if (bytesWritten == -1) {
        perror("write");
        return 1;
    }

    close(pipefd[1]);

    wait(NULL);
}

```

والد کد بالا را اجرا میکند و در ابتدا پایپ مربوط به خواندن را میبندد و مشخص میکند که میخواهد بنویسد.

سپس `message` را در `file descriptor` دوم (پایپ دوم مربوط به نوشتن میباشد) با توجه به اندازه ی داده شده (کل پیام) مینویسد و خروجی (تعداد بایت نوشته شده) در صورتی که منفی یک باشد یعنی با ارور مواجه شده و از برنامه خارج میشود. در انتها پایپ مربوط به نوشتن را میبندد و قبل از `terminate` شدن، منتظر فرزند می ماند تا `terminate` شود