

## Code 7

```
#include <stdio.h>

#include <unistd.h>

#include <string.h>

int main() {

    int fd[2];          // fd[0] = read end, fd[1] = write end

    pid_t pid;

    char write_msg[] = "Hello Child, this is Parent!";

    char read_msg[100];

    // Create pipe

    if (pipe(fd) == -1) {

        printf("Pipe failed!\n");

        return 1;

    }

    // Create child process

    pid = fork();

    if (pid < 0) {

        printf("Fork failed!\n");

        return 1;

    }

    // Parent process

    if (pid > 0) {

        close(fd[0]); // Close unused read end
```

```
    write(fd[1], write_msg, strlen(write_msg) + 1);
    close(fd[1]); // Close write end after writing
}
// Child process
else {
    close(fd[1]); // Close unused write end
    read(fd[0], read_msg, sizeof(read_msg));
    printf("Child received message: %s\n", read_msg);
    close(fd[0]); // Close read end after reading
}

return 0;
}
```