به نام خدا

دانشکده مهندسی کامپیوتر

ازمایشگاه سیستم های عامل
ازمایش هفتم: لوله (Pipe)

اعضای گروه:
حسین تاتار – 40133014
محمد امین فرح بخش – 40131029

اردیبهشت 1404

در این آزمایش، دو فرآیند (والد و فرزند) از طریق **دو لوله** (Pipe)با یکدیگر ارتباط برقرار می‌کنند.

- فرآیند والد یک پیام می‌فرستد، فرزند آن را دریافت کرده، حروف را معکوس (بزرگ به کوچک و کوچک به بزرگ) می‌کند و دوباره به والد برمی‌گرداند.
- برای این کار به **دو لوله** نیاز داریم:
  - **لوله ۱** :ارسال از والد به فرزند
  - **لوله ۲** :ارسال از فرزند به والد

*** عکس کد این بخش در ادامه اورده شده است:

Open pipeCode.c ~/Desktop

```c
#include <stdio.h>      // Standard I/O functions (printf, perror)
#include <unistd.h>     // POSIX API (pipe, fork, read, write, close)
#include <string.h>     // String manipulation (strlen)
#include <ctype.h>      // Character handling (isupper, tolower, etc.)
#include <sys/wait.h>   // Process waiting (wait)

#define BUFFER_SIZE 1024  // Buffer size for messages

// Converts uppercase letters to lowercase and vice versa in a string.
void convertCase(char *str) {
    for (int i = 0; str[i]; i++) {
        if (isupper(str[i]))
            str[i] = tolower(str[i]);
        else if (islower(str[i]))
            str[i] = toupper(str[i]);
    }
}

int main() {
    int pipe1[2];               // Pipe 1: Parent → Child (parent writes, child reads)
    int pipe2[2];               // Pipe 2: Child → Parent (child writes, parent reads)
    char buffer[BUFFER_SIZE];   // Buffer for message exchange
    pid_t pid;                  // Process ID for fork()

    // Create pipes (check for errors)
    if (pipe(pipe1) == -1 || pipe(pipe2) == -1) {
        perror("Pipe creation failed");
        return 1;
    }

    pid = fork();           // Create child process

    if (pid < 0) {          // Fork failed
        perror("Fork failed");
        return 1;
    }

    if (pid > 0) {          // Parent process
        close(pipe1[0]);  // Close unused read end of Pipe 1
        close(pipe2[1]);  // Close unused write end of Pipe 2

        char message[] = "This Is First Process";
        printf("Parent (sending): %s\n", message);

        // Send message to child via Pipe 1
        write(pipe1[1], message, strlen(message) + 1);
        close(pipe1[1]);  // Close write end after sending
```

```c
        // Wait for response from child via Pipe 2
        read(pipe2[0], buffer, BUFFER_SIZE);
        printf("Parent (received): %s\n", buffer);
        close(pipe2[0]);  // Close read end after receiving

        wait(NULL);  // Wait for child to terminate
    }
    else {  // Child process
        close(pipe1[1]);  // Close unused write end of Pipe 1
        close(pipe2[0]);  // Close unused read end of Pipe 2

        // Receive message from parent via Pipe 1
        read(pipe1[0], buffer, BUFFER_SIZE);
        printf("Child (received): %s\n", buffer);

        // Convert letter cases (uppercase ↔ lowercase)
        convertCase(buffer);

        // Send modified message back to parent via Pipe 2
        write(pipe2[1], buffer, strlen(buffer) + 1);
        close(pipe2[1]);  // Close write end after sending
        close(pipe1[0]);  // Close read end after receiving
    }
    return 0;
}
```

توضیحات خط به خط

1. ایجاد لوله‌ها :

pipe(pipe1) و pipe(pipe2)دو لوله ایجاد می‌کنند. هر لوله دو fd (File Descriptor) دارد:

- pipeX[0] برای خواندن.(Read)
- pipeX[1] برای نوشتن.(Write)

2. فراخوانی fork() :

فرآیند فرزند (pid == 0) و والد (pid > 0) ایجاد می‌شوند.

3. بخش والد :

پیام "This Is First Process"را از طریق pipe1[1] می‌نویسد. سپس پاسخ را از pipe2[0] می‌خواند و درخروجی چاپ میکند.

4. بخش فرزند :

پیام را از pipe1[0] می‌خواند. با تابع convertCase() حروف پیام دریافتی را معکوس می‌کند و نتیجه را از طریق pipe2[1] به والد برمی‌گرداند.

5. بستن لوله‌ها :

توصیفگرهای غیرضروری در هر فرآیند بسته می‌شوند تا از Deadlock جلوگیری شود.

6. تابع convertCase() :

حروف بزرگ را به کوچک (tolower) و حروف کوچک را به بزرگ (toupper) تبدیل می‌کند.

خروجی این کد به صورت زیر است :

```
hosseintatar@hosseintatar-VMware-Virtual-Platform: ~/Desktop
hosseintatar@hosseintatar-VMware-Virtual-Platform:~/Desktop$ touch pipeCode.c
hosseintatar@hosseintatar-VMware-Virtual-Platform:~/Desktop$ gcc pipeCode.c -o simulate
hosseintatar@hosseintatar-VMware-Virtual-Platform:~/Desktop$ ./simulate
Parent (sending): This Is First Process
Child (received): This Is First Process
Parent (received): tHIS iS fIRST pROCESS
hosseintatar@hosseintatar-VMware-Virtual-Platform:~/Desktop$
```

همانطور که مشاهده میشود حروف پیام ارسالی از والد در سمت فزرند تغییر اندازه داده شده و به والد فرستاده شده است.