

ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



BÁO CÁO

Bài tập thực hành :

Chương 4: Socket APIs

Học phần: Thực hành Lập Trình Mạng

Giảng viên hướng dẫn : Trần Hải Anh

Mã lớp : 151907

Sinh viên thực hiện : Phạm Văn Anh

Mã số sinh viên : 20214988

Hà Nội, tháng 9 năm 2024

Bài 1: Xử lý tiến trình zombie

Yêu cầu 1: Yêu cầu 1: Hãy viết một chương trình đơn giản sử dụng hàm fork() để tạo tiến trình con mà không dùng trình xử lý tín hiệu (signal handler). Trong đó tiến trình cha tạo ra một tiến trình con bằng cách sử dụng fork(). Tiến trình cha gọi hàm sleep(30); để ngủ 30s và không chờ tiến trình con kết thúc, điều này có thể dẫn đến tiến trình con trở thành zombie.

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

int main() {
    pid_t pid = fork();

    if (pid < 0) {
        fprintf(stderr, "Fork failed\n");
        return 1;
    } else if (pid == 0) {
        printf("This is the child process. My PID is %d.\n", getpid());
        exit(0);
    } else {
        printf("This is the parent process. My child's PID is %d.\n",
pid);
        printf("My PID is %d.\n", getpid());
        wait(NULL);
        printf("Parent process has collected the zombie process.\n");
        sleep(30);
    }

    return 0;
}
```

Câu hỏi 1: Kết quả thu được là gì? Tiến trình nào đang ở trạng thái Z? Từ khóa nghĩa là gì?

```
vaah@vaah-VirtualBox:~/Documents/lab4$ gcc parent.c -o parent
vaah@vaah-VirtualBox:~/Documents/lab4$ ./parent
This is the parent process. My child's PID is 5037.
My PID is 5036.
This is the child process. My PID is 5037.

vaah@vaah-VirtualBox:~/Documents/lab4$ ps aux | grep Z
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
vaah      5283  0.0  0.0   9144  2176 pts/2    S+   02:08   0:00 grep --color=auto Z
vaah@vaah-VirtualBox:~/Documents/lab4$
```

- **Từ khóa <defunct> nghĩa là gì:** <defunct> là từ dùng để chỉ các tiến trình đã kết thúc nhưng vẫn còn tồn tại trong bảng tiến trình của hệ điều hành. Tiến trình đó trở thành zombie vì tiến trình cha chưa thu dọn nó bằng cách gọi wait() hoặc waitpid().

Yêu cầu 2: Làm lại các bước ở trên và cho biết có còn tiến trình zombie nào xuất hiện không? vì sao?

```

vaah@vaah-VirtualBox:~/Documents/lab4$ ./parent
This is the parent process. My child's PID is 4866.
My PID is 4865.
This is the child process. My PID is 4866.
Parent process has collected the zombie process.

```

```

Parent process has collected the zombie process.
vaah@vaah-VirtualBox:~/Documents/lab4$ ps aux | grep Z
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
vaah      4978  0.0  0.0   9144  2176 pts/2    S+   02:02   0:00 grep --color=auto Z

```

- **Kết quả:** Sau khi sửa đổi và thêm wait(), khi chạy lại chương trình và kiểm tra bằng lệnh ps aux | grep Z, sẽ không còn xuất hiện bất kỳ tiến trình zombie nào nữa.
- **Giải thích:** Việc sử dụng wait() trong tiến trình cha sẽ giúp tiến trình cha thu gom tiến trình con sau khi nó kết thúc, ngăn chặn tiến trình con trở thành zombie.

Bài 2: Viết một chương trình client-server

- Server là một chương trình đa tiến trình, có thể nhận kết nối với nhiều client cùng lúc, với mỗi client thì server sẽ dùng hàm `fork()` để sinh ra một tiến trình con để trao đổi thông tin với client đó. Server có kiểm soát thu hồi tiến trình zombie client.
- Server có lưu trữ 10 câu hỏi trắc nghiệm, mỗi câu hỏi trắc nghiệm có 4 đáp án, trong đó có một đáp án đúng.
- Sau khi một client liên kết với server thì sẽ nhận được lần lượt các câu hỏi kèm theo 4 câu trả lời theo thứ tự ngẫu nhiên.
- Client nhận được câu hỏi và danh sách 4 câu trả lời thì sẽ cho người dùng gõ vào câu trả lời và gửi lại cho server. Mỗi câu trả lời đúng thì client đó được 1 điểm.
- Sau khi trả lời xong 10 câu hỏi thì chương trình sẽ gửi lại điểm của client trong lần chơi đó.

- Server:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <arpa/inet.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <signal.h>
#include <sys/wait.h>

#define PORT 8080
#define BUFFER_SIZE 1024
#define MAX_QUESTIONS 10

char *questions[MAX_QUESTIONS] = {
    "1 + 1 = ?\nA. 1\nB. 2\nC. 3\nD. 4",
    "What is the capital of France?\nA. London\nB. Paris\nC. Berlin\nD. Madrid",
    "5 * 5 = ?\nA. 20\nB. 25\nC. 30\nD. 35",
    "What is the largest planet?\nA. Earth\nB. Mars\nC. Jupiter\nD. Venus",
    "What is the boiling point of water?\nA. 90°C\nB. 100°C\nC. 110°C\nD. 120°C",
    "Which language is used for web development?\nA. Python\nB. HTML\nC. C++\nD. Java",
```

```

    "Who developed the theory of relativity?\nA. Newton\nB.
Einstein\nC. Galileo\nD. Bohr",
    "What is the capital of Japan?\nA. Seoul\nB. Tokyo\nC. Kyoto\nD.
Osaka",
    "What is 10 / 2 = ?\nA. 3\nB. 5\nC. 7\nD. 9",
    "Which element is denoted by 'O' in chemistry?\nA. Oxygen\nB.
Hydrogen\nC. Helium\nD. Carbon"
};

char correct_answers[MAX_QUESTIONS] = {'B', 'B', 'B', 'C', 'B', 'B',
'B', 'B', 'B', 'A'};

// Signal handler to prevent zombie processes
void sigchld_handler(int sig) {
    (void) sig;
    while (waitpid(-1, NULL, WNOHANG) > 0);
}

void handle_client(int connfd) {
    char buffer[BUFFER_SIZE];
    int score = 0;

    for (int i = 0; i < MAX_QUESTIONS; i++) {
        if (questions[i] != NULL) {
            send(connfd, questions[i], strlen(questions[i]), 0);
        } else {
            snprintf(buffer, sizeof(buffer), "Question %d does not
exist.\n", i+1);
            send(connfd, buffer, strlen(buffer), 0);
        }

        memset(buffer, 0, BUFFER_SIZE);
        read(connfd, buffer, sizeof(buffer));

        if (buffer[0] == correct_answers[i]) {
            score++;
        }
    }

    snprintf(buffer, sizeof(buffer), "Your final score is: %d/%d\n",
score, MAX_QUESTIONS);
    send(connfd, buffer, strlen(buffer), 0);

    printf("Client's score: %d\n", score);

    close(connfd);
    exit(0);
}

```

```

int main() {
    int listenfd, connfd;
    struct sockaddr_in server_addr, client_addr;
    socklen_t client_len = sizeof(client_addr);
    pid_t pid;

    // Create a socket
    if ((listenfd = socket(AF_INET, SOCK_STREAM, 0)) == 0) {
        perror("Socket failed");
        exit(EXIT_FAILURE);
    }

    // Define server address
    server_addr.sin_family = AF_INET;
    server_addr.sin_addr.s_addr = INADDR_ANY;
    server_addr.sin_port = htons(PORT);

    // Bind the socket
    if (bind(listenfd, (struct sockaddr *)&server_addr,
sizeof(server_addr)) < 0) {
        perror("Bind failed");
        close(listenfd);
        exit(EXIT_FAILURE);
    }

    // Listen for incoming connections
    if (listen(listenfd, 5) < 0) {
        perror("Listen failed");
        close(listenfd);
        exit(EXIT_FAILURE);
    }

    // Handle SIGCHLD to prevent zombie processes
    signal(SIGCHLD, sigchld_handler);

    printf("Server is running on port %d...\n", PORT);

    while (1) {
        // Accept incoming connection
        if ((connfd = accept(listenfd, (struct sockaddr *)&client_addr,
&client_len)) < 0) {
            perror("Accept failed");
            continue;
        }

        // Fork a child process to handle the client
        if ((pid = fork()) == 0) {

```

```

        close(listenfd);
        handle_client(connfd);
    } else if (pid > 0) {

        close(connfd);
    } else {

        perror("Fork failed");
        close(connfd);
    }
}

return 0;
}

```

- Client

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <arpa/inet.h>

#define PORT 8080
#define BUFFER_SIZE 1024
#define MAX_QUESTIONS 10

int main() {
    int client_socket;
    struct sockaddr_in server_addr;
    char buffer[BUFFER_SIZE];
    int n;

    // Create socket
    if ((client_socket = socket(AF_INET, SOCK_STREAM, 0)) < 0) {
        perror("Socket creation failed");
        exit(EXIT_FAILURE);
    }

    // Define server address
    server_addr.sin_family = AF_INET;
    server_addr.sin_port = htons(PORT);
    if (inet_pton(AF_INET, "127.0.0.1", &server_addr.sin_addr) <= 0) {
        perror("Invalid address or Address not supported");
        close(client_socket);
    }
}

```

```

        exit(EXIT_FAILURE);
    }

    // Connect to server
    if (connect(client_socket, (struct sockaddr *)&server_addr,
sizeof(server_addr)) < 0) {
        perror("Connection failed");
        close(client_socket);
        exit(EXIT_FAILURE);
    }

    for (int i = 0; i < MAX_QUESTIONS; i++) {

        memset(buffer, 0, BUFFER_SIZE);
        read(client_socket, buffer, sizeof(buffer));
        printf("Question %d: %s\n", i + 1, buffer);

        while (1) {

            printf("Enter your answer (A, B, C, or D): ");
            fgets(buffer, BUFFER_SIZE, stdin);

            if (buffer[strlen(buffer) - 1] == '\n') {
                buffer[strlen(buffer) - 1] = '\0';
            }

            if (buffer[0] == 'A' || buffer[0] == 'B' || buffer[0] ==
'C' || buffer[0] == 'D') {
                send(client_socket, buffer, strlen(buffer), 0);
                break;
            } else {
                printf("Invalid answer. Please enter A, B, C, or
D.\n");
            }
        }
    }

    // Receive and print the final score
    memset(buffer, 0, BUFFER_SIZE);
    read(client_socket, buffer, sizeof(buffer));
    printf("%s", buffer);

    close(client_socket);

    return 0;
}

```


- Client 1:

```
Question 6: Which language is used for web development?
A. Python
B. HTML
C. C++
D. Java
Enter your answer (A, B, C, or D): B
Question 7: Who developed the theory of relativity?
A. Newton
B. Einstein
C. Galileo
D. Bohr
Enter your answer (A, B, C, or D): B
Question 8: What is the capital of Japan?
A. Seoul
B. Tokyo
C. Kyoto
D. Osaka
Enter your answer (A, B, C, or D): B
Question 9: What is 10 / 2 = ?
A. 3
B. 5
C. 7
D. 9
Enter your answer (A, B, C, or D): B
Question 10: Which element is denoted by 'O' in chemistry?
A. Oxygen
B. Hydrogen
C. Helium
D. Carbon
Enter your answer (A, B, C, or D): B
Your final score is: 8/10
vaah@vaah-VirtualBox:~/Documents/lab4$
```

- Client 2:

```
C. Berlin
D. Madrid
Enter your answer (A, B, C, or D): B
Question 3: 5 * 5 = ?
A. 20
B. 25
C. 30
D. 35
Enter your answer (A, B, C, or D): B
Question 4: What is the largest planet?
A. Earth
B. Mars
C. Jupiter
D. Venus
Enter your answer (A, B, C, or D): B
Question 5: What is the boiling point of water?
A. 90°C
B. 100°C
C. 110°C
D. 120°C
Enter your answer (A, B, C, or D): B
Question 6: Which language is used for web development?
A. Python
B. HTML
C. C++
D. Java
Enter your answer (A, B, C, or D): B
Question 7: Who developed the theory of relativity?
A. Newton
B. Einstein
C. Galileo
D. Bohr
Enter your answer (A, B, C, or D): B
Question 8: What is the capital of Japan?
A. Seoul
B. Tokyo
C. Kyoto
```

- Server sau khi hoàn thành câu hỏi:

```
vaah@vaah-VirtualBox:~/Documents/lab4$ ./server
Server is running on port 8080...
Client's score: 8
█
```