

Programowanie systemowe

Sprawozdanie

Aron Krajda - 283874

Data wykonania sprawozdania: 16.12.2025

Cele:

Celem laboratorium była kolejna modyfikacja plików użytych w poprzednich laboratoriach.

Dodane zostały następujące funkcje:

- usunięto korzystanie z pipe i zastąpiono go używaniem kolejek nazwanych
- stworzono możliwość uruchomienia jednego z dwóch procesów (server lub worker, które w poprzednich laboratoriach były nazwane parent i child)

Plik main.c:

Plik main.c został zmodyfikowany tak, aby używał wpisanego przy jego uruchamianiu słowa "server" lub "worker" do decyzji o tym, czy uruchomiony zostanie proces worker czy server.

Aby program działał poprawnie muszą zostać uruchomione oba procesy.

```
#include <stdio.h>
#include <string.h>
#include "server.h"
#include "worker.h"

int main(int argc, char *argv[]) {
    if (argc != 2) {
        printf("Użycie: %s [server|worker]\n", argv[0]);
        return 1;
    }

    if (strcmp(argv[1], "server") == 0) {
        server();
    } else if (strcmp(argv[1], "worker") == 0) {
        worker();
    } else {
        printf("Użyj: server lub worker\n");
        return 1;
    }

    return 0;
}
```

Plik server.c:

Plik server.c zmodyfikowano tak, aby zamiast pipe używał mkfifo. Pozostałe funkcje pozostają takie same jak w poprzednim laboratorium.

```
#include <stdio.h>
#include <string.h>
#include <fcntl.h>
#include <unistd.h>
#include <sys/stat.h>

void server() {
    char input[256];
    char result[256];

    mkfifo("/tmp/to_worker", 0666);
    mkfifo("/tmp/to_server", 0666);

    int out = open("/tmp/to_worker", O_WRONLY);
    int in = open("/tmp/to_server", O_RDONLY);

    while (1) {
        scanf("%s", input);
        write(out, input, strlen(input) + 1);

        if (strcmp(input, "exit") == 0) break;

        read(in, result, 256);
        printf("%s\n", result);
    }

    close(out);
    close(in);
    unlink("/tmp/to_worker");
    unlink("/tmp/to_server");
}
```

Plik worker.c:

Plik worker.c zmodyfikowano tak, aby zamiast pipe używał mkfifo. Pozostałe funkcje pozostają takie same jak w poprzednim laboratorium.

```
#include <string.h>
#include <fcntl.h>
#include <unistd.h>

void worker() {
    char input[256];

    int in = open("/tmp/to_worker", O_RDONLY);
    int out = open("/tmp/to_server", O_WRONLY);

    while (1) {
        read(in, input, 256);
        if (strcmp(input, "exit") == 0) break;

        input[0] = 'X';
        write(out, input, strlen(input) + 1);
    }

    close(in);
    close(out);
}
```

Uruchomienie i wnioski:

Po uruchomieniu obu procesów program działa tak, jak działał poprzednio, co wskazuje na poprawne wprowadzenie zmian. Poprawnie zaimplementowano mkfifo oraz możliwość uruchomienia ręcznie procesu server oraz worker:

```
vboxuser@ubuntu22: ~/student3

vboxuser@ubuntu22:~$ cd student 3
bash: cd: too many arguments
vboxuser@ubuntu22:~$ cd student3
vboxuser@ubuntu22:~/student3$ ./program worker
vboxuser@ubuntu22:~/student3$ █

vboxuser@ubuntu22: ~/student3

vboxuser@ubuntu22:~$ cd student3
vboxuser@ubuntu22:~/student3$ make
make: Nothing to be done for 'all'.
vboxuser@ubuntu22:~/student3$ ./program server
adsda
Xdsda
sa
Xa
exit
vboxuser@ubuntu22:~/student3$ █
```