Zad 1

Kod klienta:

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <sys/types.h>

#include <sys/stat.h>

#include <fcntl.h>

#include <string.h>

#include <sys/wait.h>

#define FIFO\_SERVER "fifo\_serwer"

int main(int argc, char \*argv[]) {

int num\_processes = atoi(argv[1]);

for (int i = 0; i < num\_processes; ++i) {

pid\_t pid = fork();

if (pid == -1) {

perror("Błąd fork");

exit(EXIT\_FAILURE);

} else if (pid == 0) {

char fifo\_client[20]; // Adjust the size as needed

snprintf(fifo\_client, sizeof(fifo\_client), "fifo\_pid\_%d", getpid());

// tworzenie kolejki fifo

if (mkfifo(fifo\_client, 0666) == -1) {

perror("Błąd fifo");

exit(EXIT\_FAILURE);

}

// zapisywanie kolejki

int fd\_server = open(FIFO\_SERVER, O\_WRONLY);

if (fd\_server == -1) {

perror("Błąd odwarcia FIFO serwera");

exit(EXIT\_FAILURE);

}

write(fd\_server, fifo\_client, strlen(fifo\_client) + 1);

close(fd\_server);

// otwarcie odczytywania

int fd\_client = open(fifo\_client, O\_RDONLY);

if (fd\_client == -1) {

perror("Błąd otwarcia FIFO klient");

exit(EXIT\_FAILURE);

}

// drukowanie ls

char buffer[100];

while (read(fd\_client, buffer, sizeof(buffer)) > 0) {

printf("Klient %d: %s", getpid(), buffer);

}

// zamykanie kolejki

close(fd\_client);

unlink(fifo\_client);

exit(EXIT\_SUCCESS);

}

}

// czekanie na potomnych

for (int i = 0; i < num\_processes; ++i) {

wait(NULL);

}

return 0;

}

Kod serwera:

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <sys/types.h>

#include <sys/stat.h>

#include <fcntl.h>

#include <string.h>

#define FIFO\_SERVER "fifo\_serwer"

int main() {

// tworzenie kolejki

if (mkfifo(FIFO\_SERVER, 0666) == -1) {

perror("Błąd tworznia kolejki");

exit(EXIT\_FAILURE);

}

// Otwórz odczyt FIFO

int fd\_server = open(FIFO\_SERVER, O\_RDONLY);

if (fd\_server == -1) {

perror("Błąd odczyt FIFO serwer");

exit(EXIT\_FAILURE);

}

// Czekanie na klientow

while (1) {

char fifo\_client[256];

// czytanie nazwy klienta

if (read(fd\_server, fifo\_client, sizeof(fifo\_client)) > 0) {

// Otwarcie FIFO zapis

int fd\_client = open(fifo\_client, O\_WRONLY);

if (fd\_client == -1) {

perror("Błąd otwarcia FIFO klienta");

continue;

}

// Wykonaj polecenie ls i przekaż wynik do klienta

FILE \*ls\_output = popen("ls", "r");

char buffer[256];

while (fgets(buffer, sizeof(buffer), ls\_output) != NULL) {

write(fd\_client, buffer, strlen(buffer));

}

// Zamkniecie polaczenia

pclose(ls\_output);

close(fd\_client);

printf("Zapis do %s\n", fifo\_client);

}

}

close(fd\_server);

unlink(FIFO\_SERVER);

return 0;

}

A screenshot of a computer screen

Description automatically generated