

Programowanie systemowe

Sprawozdanie

Aron Krajda - 283874

Data wykonania sprawozdania: 24.11.2025

Cele:

Celem zadania było napisanie programu w C, który:

- wczyta tekst od użytkownika
- zmieni pierwszą literę tego tekstu na 'X'
- wyświetli zmieniony tekst
- podzieli poprzednie działania na 2 procesy przy użyciu fork, tak aby proces rodzic wczytywał tekst, a proces dziecko zmieniał pierwszą literę i wyświetlał nowy tekst
- użyje pipe do komunikacji między procesami

Plik zadanie.c:

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

int main() {
    char input[256];
    int fd[2];
    char input2[256];
    pipe(fd);
    pid_t id = fork();

    if (id > 0) {
        close(fd[0]);
        scanf("%s", input);
        write(fd[1], input, strlen(input)+1);
        close(fd[1]);
        wait(NULL);
    }

    if (id == 0) {
        close(fd[1]);
        read(fd[0], input2, 256);
        close(fd[0]);

        input2[0] = 'X';

        printf("%s", input2);
    }

    return 0;
}
```

Pierwsze 3 linijki pliku zawierają nagłówki służące do dodania: funkcji wejścia/wyjścia, funkcji na stringach oraz funkcji systemowych.

Następnie w funkcji main znajdują się definicje zmiennych i utworzenie pipe.

Zmienne to:

- input[256] - tablica na wczytany tekst
- fd[2] - deskryptory pipe
- input2[256] - tablica na nowy tekst

Następne dwie linijki służą do utworzenia pipe

Następny fragment wykonywany jest przez proces rodzica (czyli gdy $id > 0$). Wczytuje on tekst i wysyła do procesu dziecka, a później czeka na zakończenie procesu dziecka.

Kolejny fragment wykonywany jest przez proces dziecko (czyli gdy $id = 0$). Odbiera on dane przez pipe, zmienia pierwszą literę, a następnie wyświetla zmodyfikowany tekst.

Następnie kończy się działanie programu, ponieważ wszystkie warunki zadania zostały spełnione.

Uruchomienie programu:

```
Hello vboxuser@ubuntu22:~/Downloads$ gcc zadanie.c -o zadanie
zadanie.c: In function 'main':
zadanie.c:17:17: warning: implicit declaration of function 'wait' [-Wimplicit-fun
nction-declaration]
    17 |             wait(NULL);
        |             ^~~~~
vboxuser@ubuntu22:~/Downloads$ ./zadanie
Hello
Xello vboxuser@ubuntu22:~/Downloads$
```

Po uruchomieniu programu pierwsza litera wczytanego tekstu zostaje zmieniona na 'X', co wskazuje na to, że program działa poprawnie.

Sprawdzenie czy program rozdziela się na 2 procesy:

```
vboxuser@ubuntu22:~$ ps aux | grep zadanie
vboxuser    5850  0.0  0.0   2680   1660 pts/0    S+   09:49   0:00  ./zadanie
vboxuser    5851  0.0  0.0   2548    348 pts/0    S+   09:49   0:00  ./zadanie
vboxuser    5877  0.0  0.0   9144   2268 pts/2    S+   09:50   0:00  grep --color=
auto zadanie
vboxuser@ubuntu22:~$
```

Po wydaniu widocznej komendy, wynik wskazuje na to, że program rozdziela się na 2 procesy, co wskazuje na poprawne wykonanie zadania.

Wnioski:

Program wykonuje założone zadania, czyli wczytanie tekstu od użytkownika w procesie rodzica, przesłanie tekstu przez pipe do procesu dziecka, który to zmienia pierwszą literę na

literę 'X' oraz wyświetla tekst. Poprawne wykonanie zadania wskazują poprzednie dwa zrzuty ekranu.