

INSTYTUT TELEINFORMATYKI I AUTOMATYKI

Wydział Cybernetyki WAT

Przedmiot: SYSTEMY OPERACYJNE

SPRAWOZDANIE Z ĆWICZENIA LABORATORYJNEGO Nr 8

Temat ćwiczenia: ŁĄCZA NAZWANIE I NIENAZWANE

Wykonał:

Imię i Nazwisko

Grupa: I4X3S1

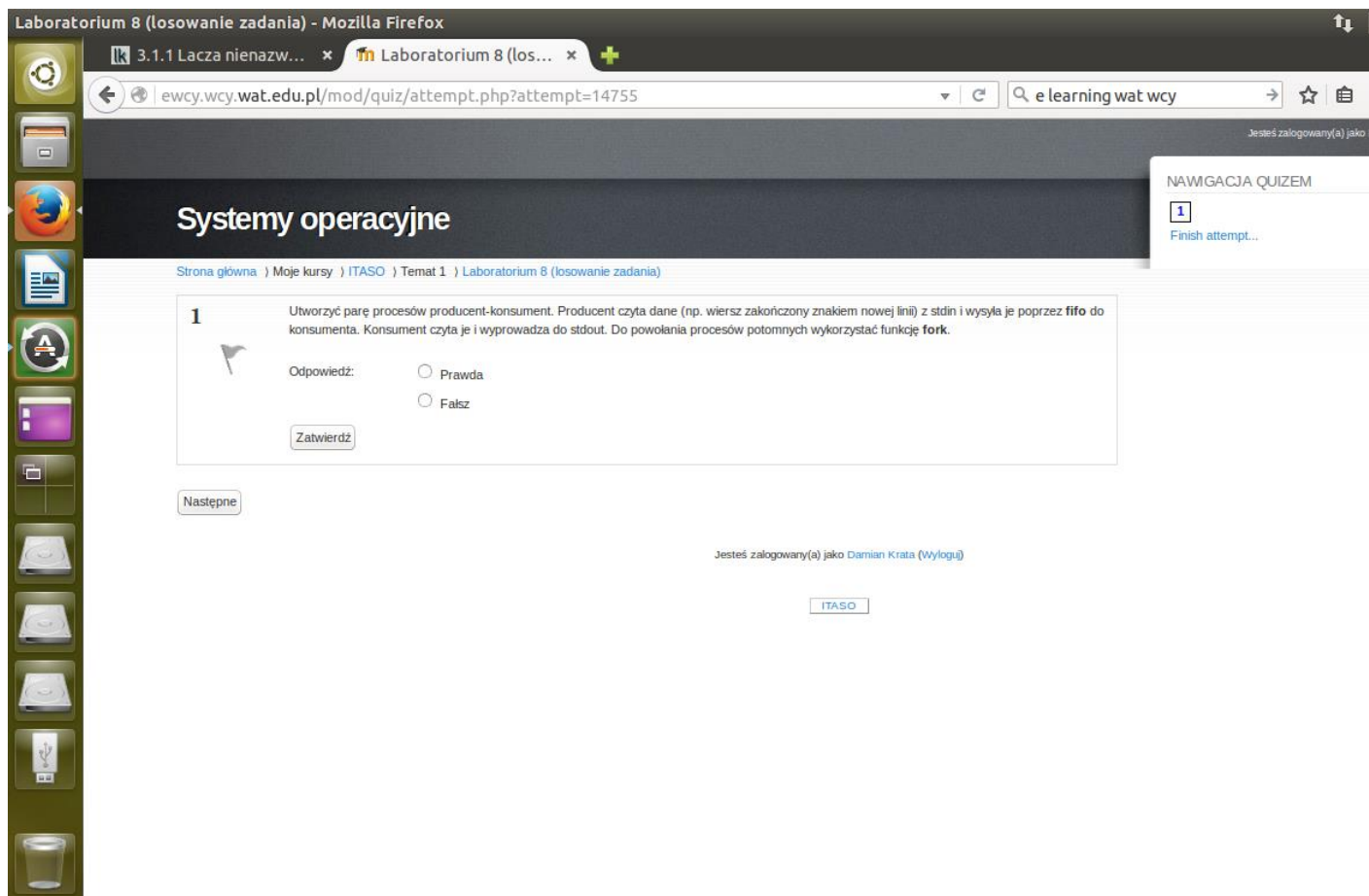
Ćwiczenie wykonane dnia
04.12.2015

Prowadzący ćwiczenie
mgr inż. Łukasz Laszko

Ocena:

.....

Zadanie:



Kod programu:

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

#define FIFO "fifoFile"

void konsument();
void producent();

int main()
{
```

```

mknod(FIFO, S_IFIFO|0666, 0);
if(fork())

        {
                producent();
        }

    else
        {
                konsument();
        }

    return 0;
}

void konsument() {
    FILE *f;
    char bufor[200];
    while(true){
        f = fopen(FIFO, "r");
        fgets(bufor, 200, f);
        printf("Odebrano : %s\n\n", bufor);
        fclose(f);
    }
}

void producent() {
    FILE *f;
    char bufor[200], *wskaznik;
    wskaznik = bufor;

    while(true){
        f = fopen(FIFO, "w");
        fprintf(stdout, "Podaj dane do wyslania : ");
        fscanf(stdin, "%s", wskaznik);
        fputs(wskaznik, f);
        fclose(f);
        sleep(2);
    }
}

```

Trochę teorii:

łącza służą do komunikacji między procesami w systemie

Linux wyróżnia się dwa rodzaje łączy:

nienazwane, inaczej potoki

nazwane, inaczej kolejki FIFO

łącza są implementowane jako pliki specjalne

posiadają swój i-węzeł

posiadają bloki z danymi

można do nich pisać i z nich czytać

Funkcja mknod()

Funkcja ta służy do utworzenia nowej kolejki FIFO.

DEFINICJA: `int mknod(nazwa_lacza, prawa_dostepu | S_IFIFO, 0)`

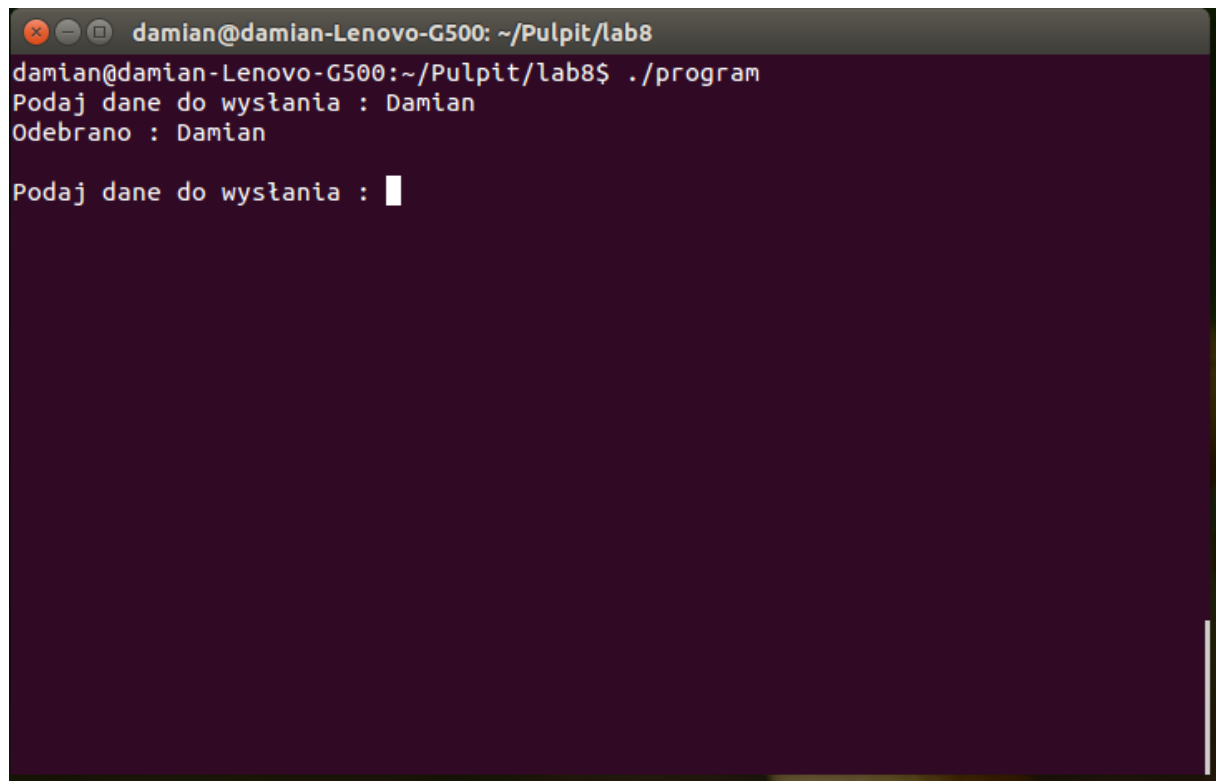
WYNIK: 0 w przypadku powodzenia operacji,

-1, gdy błąd: `errno = EEXIST` (kolejka o podanym kluczu istnieje, więc niemożliwe jest jej utworzenie)

Ponadto w moim programie deklaruję dwie funkcje: Producenta oraz konsumenta.

Używają one 200 znakowego bufora oraz zapisują i wypisują z niego.

Efekt wywołania programu:



```
damian@damian-Lenovo-G500: ~/Pulpit/lab8
damian@damian-Lenovo-G500:~/Pulpit/lab8$ ./program
Podaj dane do wysłania : Damian
Odebrano : Damian
Podaj dane do wysłania : █
```

```
damian@damian-Lenovo-G500: ~/Pulpit/lab8
damian@damian-Lenovo-G500:~/Pulpit/lab8$ ./program
Podaj dane do wysłania : Damian
Odebrano : Damian

Podaj dane do wysłania : Krata
Odebrano : Krata

Podaj dane do wysłania : █
```

```
damian@damian-Lenovo-G500: ~/Pulpit/lab8
damian@damian-Lenovo-G500:~/Pulpit/lab8$ ./program
Podaj dane do wysłania : Damian
Odebrano : Damian

Podaj dane do wysłania : Krata
Odebrano : Krata

Podaj dane do wysłania : ciekawe dlaczego to jest takie trudne
Odebrano : ciekawe

Odebrano : dlaczego

Odebrano : to

Odebrano : jest

Odebrano : takie

Odebrano : trudne

█
```

Wnioski:

Widzimy, że program czyta to co podawane jest na wejściu a następnie odczytuje to na standardowe wyjście. Ponadto dla dłuższego tekstu wypisywane są pojedyncze napisy, jest to związane z takim a nie innym odczytywaniem strumienia (tak samo jak podczas czytania z pliku kiedy to natrafiamy na białe znaki). Ćwiczenie sprawiło mi bardzo dużo trudności, ponieważ czas poświęcony na jego zrobienie mimo małej złożoności zadania był długi. Efekt działania programy jest dobry, a wiedzę przyswoiłem w stopniu zadowalającym.