Kamil Borkowski WCY22IY1S1 83374

zad 4 semafory

Kod źródłowy:

#include <stdio.h>

#include <stdlib.h>

#include <semaphore.h>

#include <unistd.h>

#include <sys/ipc.h>

#include <sys/shm.h>

sem\_t \*semafor;

sem\_t \*semPom;

sem\_t \*sem1do2;

sem\_t \*sem2do3;

sem\_t \*sem3do1;

void fn(void \*arg, int process\_id)

{

printf("Konsument %d (%d) odczytalem: %s", process\_id, getpid(), (char \*)arg);

sem\_post(semPom);//pozwolenie napisania tekstu

}

int main()

{

char \*buffer;

size\_t bufsize = 32;

size\_t characters;

pid\_t k1, k2, k3;

int sem\_shmid = shmget(IPC\_PRIVATE, sizeof(sem\_t), IPC\_CREAT | 0666);//tworzenie pamieci wspoldzielonej

int semPom\_shmid = shmget(IPC\_PRIVATE, sizeof(sem\_t), IPC\_CREAT | 0666);

int buf\_shmid = shmget(IPC\_PRIVATE, bufsize \* sizeof(char), IPC\_CREAT | 0666);

int sem1do2\_shmid = shmget(IPC\_PRIVATE, sizeof(sem\_t), IPC\_CREAT | 0666);

int sem2do3\_shmid = shmget(IPC\_PRIVATE, sizeof(sem\_t), IPC\_CREAT | 0666);

int sem3do1\_shmid = shmget(IPC\_PRIVATE, sizeof(sem\_t), IPC\_CREAT | 0666);

semafor = (sem\_t \*)shmat(sem\_shmid, NULL, 0);//przypisanie semaforow do pamieci

semPom = (sem\_t \*)shmat(semPom\_shmid, NULL, 0);

buffer = (char \*)shmat(buf\_shmid, NULL, 0);//przypisanie buffera do pamieci

sem1do2 = (sem\_t \*)shmat(sem1do2\_shmid, NULL, 0);//przypisanie semaforow pilnujacych kolejnosci do pamieci

sem2do3 = (sem\_t \*)shmat(sem2do3\_shmid, NULL, 0);

sem3do1 = (sem\_t \*)shmat(sem3do1\_shmid, NULL, 0);

sem\_init(semafor, 1, 0);//inicjalizacja semaforow

sem\_init(semPom, 1, 1);

sem\_init(sem1do2, 1, 0);//inicjalizacja semaforow pilnujacych kolejnosci

sem\_init(sem2do3, 1, 0);

sem\_init(sem3do1, 1, 1);//ten ktory ma na ostatni arguemnt jako 1 zacznie odczyt jako pierwszy

if ((k1 = fork()) == 0)

{

while(1)

{

sem\_wait(sem3do1);//czekanie az k3 odczyta tekst

for(int i=0;i<4;i++)

{

sem\_wait(semafor);//czekanie na napisanie tekstu

fn((void \*)buffer, 1);

}

sem\_post(sem1do2);//pozwolenie k2 na odczytanie tekstu

}

}

if ((k2 = fork()) == 0)

{

while(1)

{

sem\_wait(sem1do2);//czekanie az k1 odczyta tekst

for(int i=0;i<2;i++)

{

sem\_wait(semafor);//czekanie na napisanie tekstu

fn((void \*)buffer, 2);

}

sem\_post(sem2do3);//pozwolenie k3 na odczytanie tekstu

}

}

if ((k3 = fork()) == 0)

{

while(1)

{

sem\_wait(sem2do3);//czekanie az k2 odczyta tekst

sem\_wait(semafor);//czekanie na napisanie tekstu

fn((void \*)buffer, 3);

sem\_post(sem3do1);//pozwolenie k1 na odczytanie tekstu

}

}

while (1)

{

sem\_wait(semPom);//czekanie na odczytanie tekstu

printf("Podaj tekst: ");

characters = getline(&buffer, &bufsize, stdin);

sem\_post(semafor);//pozwolenie odczytu tekstu

}

shmdt(semafor);//odlaczanie semaforow i buffera od pamieci wspodzielonej

shmdt(semPom);

shmdt(buffer);

shmdt(sem1do2);//odlaczanie semaforow pilnujacych kolejnosci od pamieci wspodzielonej

shmdt(sem2do3);

shmdt(sem3do1);

shmctl(sem\_shmid, IPC\_RMID, NULL);//usuwanie pamieci wspodzielonej

shmctl(semPom\_shmid, IPC\_RMID, NULL);

shmctl(buf\_shmid, IPC\_RMID, NULL);

shmctl(sem1do2\_shmid, IPC\_RMID, NULL);

shmctl(sem2do3\_shmid, IPC\_RMID, NULL);

shmctl(sem3do1\_shmid, IPC\_RMID, NULL);

return 0;

}

Obraz zawierający tekst, zrzut ekranu, oprogramowanie

Opis wygenerowany automatycznie