Zad 2 Sygnał

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <signal.h>

#include <sys/types.h>

#include <sys/wait.h>

#include <time.h>

void handleSIGTERM(int signum) {

printf("Usunieto proces %d\n", getpid());

exit(0);

}

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

int numChildProcesses = atoi(argv[1]);

// Ustawienie obslugi sygnału SIGTERM w procesie potomnym

signal(SIGTERM, handleSIGTERM);

// Tworzenie procesow potomnych

for (int i = 0; i < numChildProcesses; ++i) {

pid\_t childPid = fork();

if (childPid == -1) {

perror("Blad podczas fork");

return 0;

} else if (childPid == 0) {

// Proces potomny

// Losowe wygenerowanie wartości

srand(time(NULL) + getpid());

int randomValue = rand() % 10;

printf("Proces potomny %d, wartosc wylosowana: %d\n", getpid(), randomValue);

int sleepTime = randomValue;

sleep(sleepTime);

printf("Obudzil sie proces %d \n", getpid());

exit(0);

}

}

// Czekanie na zakonczenie pierwszego procesu potomnego

int status;

pid\_t firstChildPid = wait(&status);

// Zakonczenie pozostalych procesow potomnych

for (int i = 0; i < numChildProcesses - 1; ++i) {

kill(0, SIGTERM); // Wyslanie sygnalu SIGTERM do grupy procesow

wait(NULL);

}

printf("Proces macierzysty zakonczył sie.\n");

return 0;

}}A screenshot of a computer screen

Description automatically generated