Zadanie 1 nr10

#include <stdio.h>

#include <stdlib.h>

#include <sys/types.h>

#include <sys/wait.h>

#include <unistd.h>

int main() {

pid\_t pid2, pid3, pid4, pid5, pid6, pid7, pid8, pid9, pid10;

pid2 = fork();

switch (pid2) {

case 0:

// P2

pid5 = fork();

switch (pid5) {

case 0:

// P5

pid8 = fork();

switch (pid8) {

case 0:

// P8

printf("P8: Pid: %d, ppid: %d \n", getpid(), getppid());

exit(0);

case -1:

perror("P8 blad");

exit(1);

default:

wait(NULL);

}

pid9 = fork();

switch (pid9) {

case 0:

// P9

printf("P9: Pid: %d, ppid: %d \n", getpid(), getppid());

exit(0);

case -1:

perror("P9 blad");

exit(1);

default:

wait(NULL);

}

printf("P5: Pid: %d, ppid: %d, potomkowie: %d %d \n", getpid(), getppid(), pid8, pid9);

exit(0);

case -1:

perror("P5 blad");

exit(1);

default:

wait(NULL);

}

pid4 = fork();

switch (pid4) {

case 0:

// P4

printf("P4: Pid: %d, ppid: %d \n", getpid(), getppid());

exit(0);

case -1:

perror("P4 blad");

exit(1);

default:

wait(NULL);

}

printf("P2: Pid: %d, ppid: %d, potomkowie: %d %d\n", getpid(), getppid(), pid4, pid5);

break;

case -1:

perror("P2 fork error");

exit(1);

default:

// P1

pid3 = fork();

switch (pid3) {

case 0:

// P3

pid6 = fork();

switch (pid6) {

case 0:

// P6

printf("P6: Pid: %d, ppid: %d \n", getpid(), getppid());

exit(0);

case -1:

perror("P6 blad");

exit(1);

default:

wait(NULL);

}

pid7 = fork();

switch (pid7) {

case 0:

// P7

pid10 = fork();

switch (pid10) {

case 0:

// P10

printf("P10: Pid: %d, ppid: %d \n", getpid(), getppid());

exit(0);

case -1:

perror("P10 blad");

exit(1);

default:

wait(NULL);

}

printf("P7: Pid: %d, ppid: %d, potomek: %d \n", getpid(), getppid(), pid10);

exit(0);

case -1:

perror("P7 blad");

exit(1);

default:

wait(NULL);

}

printf("P3: Pid: %d, ppid: %d, potomkowie: %d %d\n", getpid(), getppid(), pid6, pid7);

exit(0);

case -1:

perror("P3 blad");

exit(1);

default:

wait(NULL);

}

wait(NULL);

printf("P1: Pid: %d, ppid: %d, potomkowie: %d %d\n", getpid(), getppid(), pid2, pid3);

}

return 0;

}

Kod posiada zagnieżdżone w switch case’ach kolejne switch case’y aby wywoływać forki w forkach, przy kończeniu switch case’a jest wypisywany rodzic z jego potomkami.

A screenshot of a computer

Description automatically generated