**Operációs rendszerek BSc**8. gyak  
2021.03.31.

**Készítette:**

Ferencsik Márk BSc

Programtervező informatika

UJTWLL

1. **ujtwll\_tobbszignal\_kez.c:**

#include <stdio.h>

#include <signal.h>

void handleSignals(int signal);

int main()

{

void (\*sigHandlerInterrupt)(int);

void (\*sigHandlerQuit)(int);

void (\*sigHandlerReturn)(int);

Szignálkezelő metúdusok deklarálása.

sigHandlerInterrupt = sigHandlerQuit = handleSignals;

sigHandlerReturn = signal(SIGINT, sigHandlerInterrupt);

if (sigHandlerReturn == SIG\_ERR)

{

perror("signal error: ");

return 1;

}

Hibás SIGINT szignál esetén kiíratás.

sigHandlerReturn = signal(SIGQUIT, sigHandlerQuit);

if (sigHandlerReturn == SIG\_ERR)

{

perror("signal error: ");

return 1;

}

Hibás SIGQUIT szignál esetén kiíratás.

while (1)

{

printf("A program leallitasahoz:\n");

printf("Nyisson meg egy masik terminalt\n");

printf("Adja ki a parancsot: kill %d vagy Control + C ketszer\n", getpid());

sleep(10);

}

return 0;

}

Program futása végtelen ciklusban.

void handleSignals(int signalNumber)

{

switch (signalNumber)

{

case SIGINT:

printf("Nyomja meg a Control + C billentyukombinaciot!\n");

signal(SIGINT, SIG\_DFL);

break;

case SIGQUIT:

printf("Nyomja meg a Control +\\\n");

break;

default:

printf("Fogadott signal szama: %d\n", signalNumber);

break;

}

}

Szignálkezelő függvény deklarációja.