#include <unistd.h>

#include <signal.h>

#include <stdlib.h>

#include <stdio.h>

int ntimes = 0;

int main() {

pid\_t pid, ppid;

void p\_action(int), c\_action(int);

static struct sigaction pact, cact;

/\* set SIGUSR1 action for parent \*/

pact.sa\_handler = p\_action;

sigaction(SIGUSR1, &pact, NULL);

switch(pid = fork()) {

case -1: /\* error \*/

perror("synchro");

exit(1);

case 0: /\* child \*/

/\* set action for child \*/

cact.sa\_handler = c\_action;

sigaction(SIGUSR1, &cact, NULL);

/\* get parent process-id \*/

ppid = getppid();

for(;;) {

sleep(1);

kill(ppid, SIGUSR1);

pause();

}

/\* never exits \*/

default: /\* parent \*/

for(;;) {

pause();

sleep(1);

kill(pid, SIGUSR1);

}

/\* never exits \*/

}

}

void p\_action(int sig) {

printf("Parent caught signal #%d\n\n", ntimes + 1);

ntimes++;

}

void c\_action(int sig) {

for (int i = 0; i < 3; i++)

printf("Child %d caught signal #%d\n", i + 1, ntimes + 1);

ntimes++;

printf("\n");

}

Text

Description automatically generatedText

Description automatically generated