



Монгол улсын их сургууль
Мэдээллийн технологи, электроникийн сургууль
Мэдээлэл, компьютерийн ухааны тэнхим

Семинарын даалгавар №11

Сэдэв: Протокол

Гүйцэтгэсэн:

Програм хангамж-3 Э.Түвшин-Эрдэнэ (23b1num0869)

Шалгасан багш:.....Д.Тэмүүжин

Улаанбаатар хот

2025 он

Дасгал 1

```
#include <stdio.h>
#include <signal.h>
#include <unistd.h>
#include <sys/wait.h>
#include <sys/types.h>
#include <stdlib.h>

volatile long counter = 2;

void handler1(int sig)
{
    sigset_t mask, prev_mask;

    sigfillset(&mask);
    sigprocmask(SIG_BLOCK, &mask, &prev_mask);

    counter--;
    printf("%ld", counter);
    fflush(stdout);

    sigprocmask(SIG_SETMASK, &prev_mask, NULL);

    _exit(0);
}

int main()
{
    pid_t pid;
    sigset_t mask, prev_mask;

    printf("%ld", counter);
    fflush(stdout);

    signal(SIGUSR1, handler1);

    if ((pid = fork()) == 0)
```

```
{
    while (1)
    {
    }

}

kill(pid, SIGUSR1);
waitpid(pid, NULL, 0);

sigfillset(&mask);
sigprocmask(SIG_BLOCK, &mask, &prev_mask);
printf("%ld", ++counter);
fflush(stdout);
sigprocmask(SIG_SETMASK, &prev_mask, NULL);

return 0;
}
```

Энэ программын үр дүнд 213 гэсэн тоо хэвлэгдэж байна.

- Эхлээд эцэг процесс counter-ийн утга болох 2-ийг хэвлэнэ.
- Хүүхэд процесс эхлэн, хязгааргүй давталт үргэлжилнэ.
- Эцэг процесс kill() функцээр хүүхэд процессыг зогсоох ба handler1-ийн дагуу counter-ийн утгыг 1-ээр багасган хэвлэнэ.
- Дараа нь counter-ийн анхны утгыг нэгээр нэмэгдүүлэн хэвлэнэ.(printf("%ld", ++counter);)

Дасгал 2

```
#include "csapp.h"
#include <signal.h>
#include <unistd.h>

void sigint_handler(int sig)
{
    sio_puts("\ninterrupted by ctrl+c\n");
}

int main()
{
    if (signal(SIGINT, sigint_handler) == SIG_ERR)
```

```
{
    sio_puts("Error installing signal handler\n");
    return 1;
}

while (1)
{
    pause(); // Wait for signals
}

return 0;
}
```

```
tuвшineerdene@pop-os:~/Documents/semester5/computer_system/seminar11$ ./exercise2
^C
interrupted by ctrl+c
```

Дасгал 3

```
#define _POSIX_C_SOURCE 200809L
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <signal.h>
#include <sys/wait.h>
#include <errno.h>
#include <string.h>

volatile sig_atomic_t children_left = 0;

static void safe_write(const char *s)
{
    ssize_t unused = write(STDOUT_FILENO, s, (size_t)strlen(s));
    (void)unused;
}

static void sigchld_handler(int signo)
{
    (void)signo;
```

```

pid_t pid;
int status;

while ((pid = waitpid(-1, &status, WNOHANG)) > 0)
{
    children_left--;

    char buf[128];
    int len = snprintf(buf, sizeof buf, "reaped child %d,
children_left=%d\n",
                                (int)pid, (int)children_left);
    if (len > 0)
    {
        write(STDOUT_FILENO, buf, (size_t)len);
    }
}

int main(int argc, char *argv[])
{
    if (argc != 2)
    {
        fprintf(stderr, "usage: %s <N>\n", argv[0]);
        return 1;
    }

    long N = strtol(argv[1], NULL, 10);
    if (N <= 0)
    {
        fprintf(stderr, "N must be positive\n");
        return 1;
    }

    sigset_t block_mask, prev_mask, empty_mask;
    sigemptyset(&block_mask);
    sigaddset(&block_mask, SIGCHLD);
    if (sigprocmask(SIG_BLOCK, &block_mask, &prev_mask) < 0)
    {
        perror("sigprocmask");
        return 1;
    }

```

```

}

struct sigaction sa;
sa.sa_handler = sigchld_handler;
sigemptyset(&sa.sa_mask);
sa.sa_flags = SA_RESTART;
if (sigaction(SIGCHLD, &sa, NULL) < 0)
{
    perror("sigaction");
    return 1;
}

children_left = (sig_atomic_t)N;

for (long i = 0; i < N; ++i)
{
    pid_t pid = fork();
    if (pid < 0)
    {
        perror("fork");
        children_left--;
        continue;
    }
    else if (pid == 0)
    {
        sleep(1);
        _exit(0);
    }
}

sigemptyset(&empty_mask);

while (children_left > 0)
{
    if (sigsuspend(&empty_mask) == -1 && errno != EINTR)
    {
        perror("sigsuspend");
        break;
    }
}

```

Компьютерийн системийн зарчим 1

```
}

printf("parent: all children reaped (children_left = %d). Exiting.\n",
(int)children_left);

if (sigprocmask(SIG_SETMASK, &prev_mask, NULL) < 0)
{
    perror("sigprocmask restore");
}

return 0;
}
```

```
• tuvshineerdene@pop-os:~/Documents/semester5/computer_system/seminar11$ ./exercise3 5
reaped child 83608, children_left=4
reaped child 83609, children_left=3
reaped child 83610, children_left=2
reaped child 83611, children_left=1
reaped child 83612, children_left=0
parent: all children reaped (children_left = 0). Exiting.
```