오픈소스리눅스프로그래밍 기말 과제

- IPC를 활용한 프로그래밍-

최서윤 (seoyun@pel.smuc.ac.kr)

상명대학교 컴퓨터공학과 201621173

목 차

- 개요
- 아키텍처
- 다이어그램
- 코드 설명

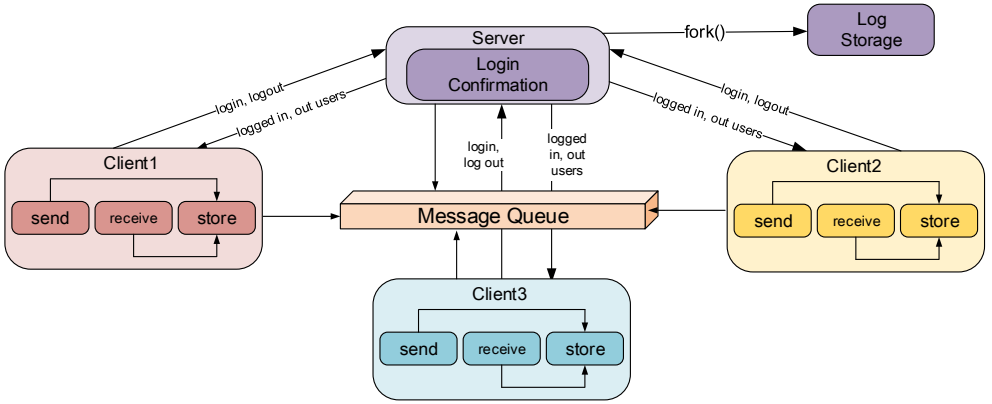
개요

- 주제
  - 부모-자식 프로세스 간의 메시지 큐 통신

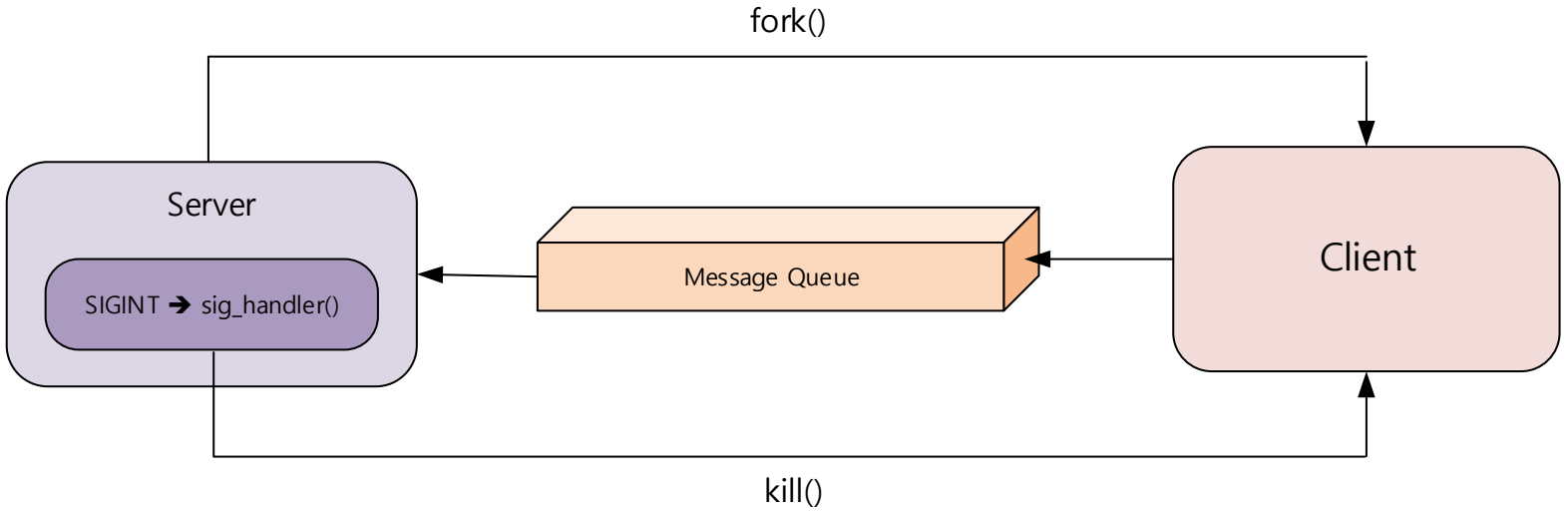
- 기술 설정
  - IPC 관련 기술
    - PIPE, FIFO, 공유메모리, 메시지 큐, 시그널, 세마포어
  - 기타 기술 활용
    - fork(), pthread(), mutex, Unix Domain Socket…

아키텍처

- 꿈…


Server
Login Confirmation
fork()
Log Storage
login, logout
logged in, out users
login, log out
logged in, out users
Client1
send
receive
store
Message Queue
Client2
send
receive
store
Client3
send
receive
store

- 현실

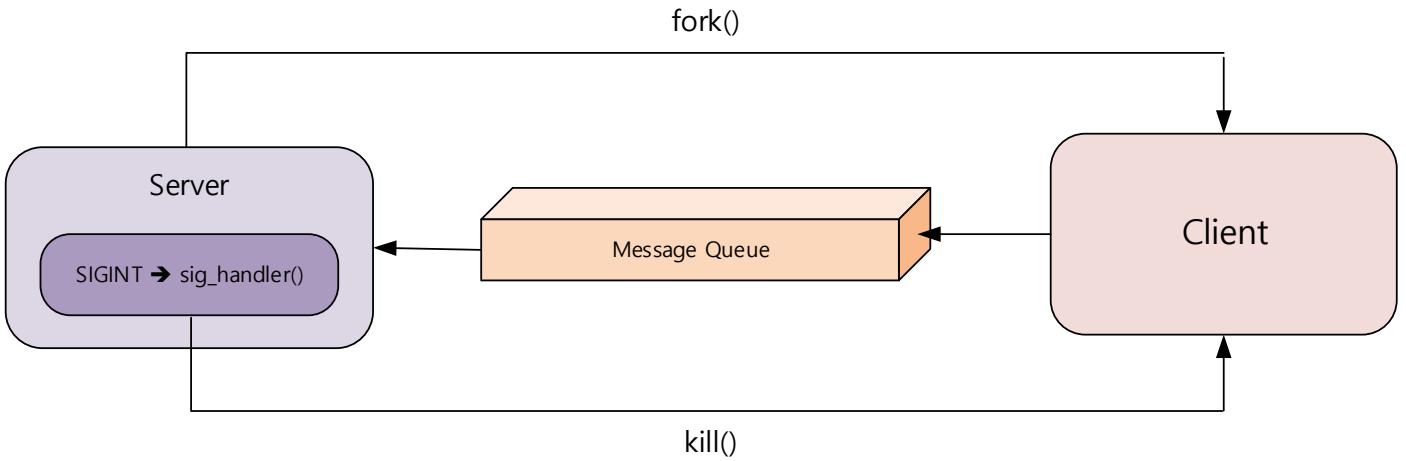

fork()
Server
SIGINT ➔ sig_handler()
Message Queue
Client
kill()

아키텍처

- fork()를 통해 자식 프로세스 생성
  - 자식 프로세스는 메시지 큐를 통해 메시지 송신
  - 부모 프로세스는 메시지 큐를 통해 메시지 수신

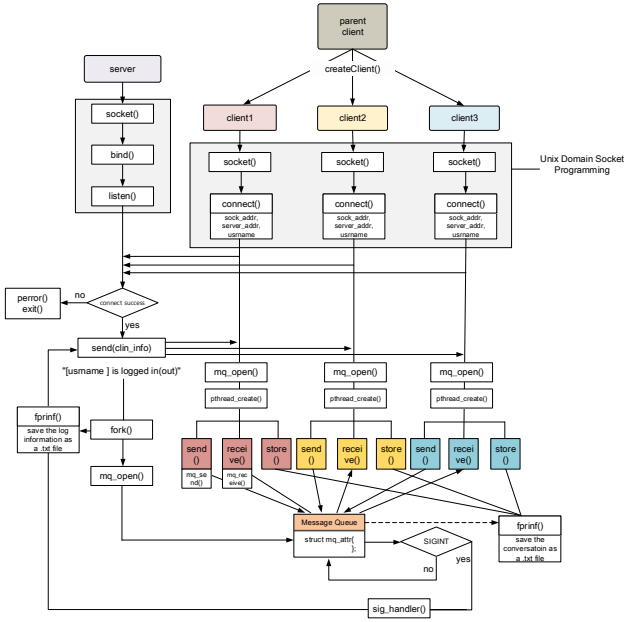
- 메시지 큐 생성

- SIGINT 입력 시 부모 프로세스와 자식 프로세스 모두 종료

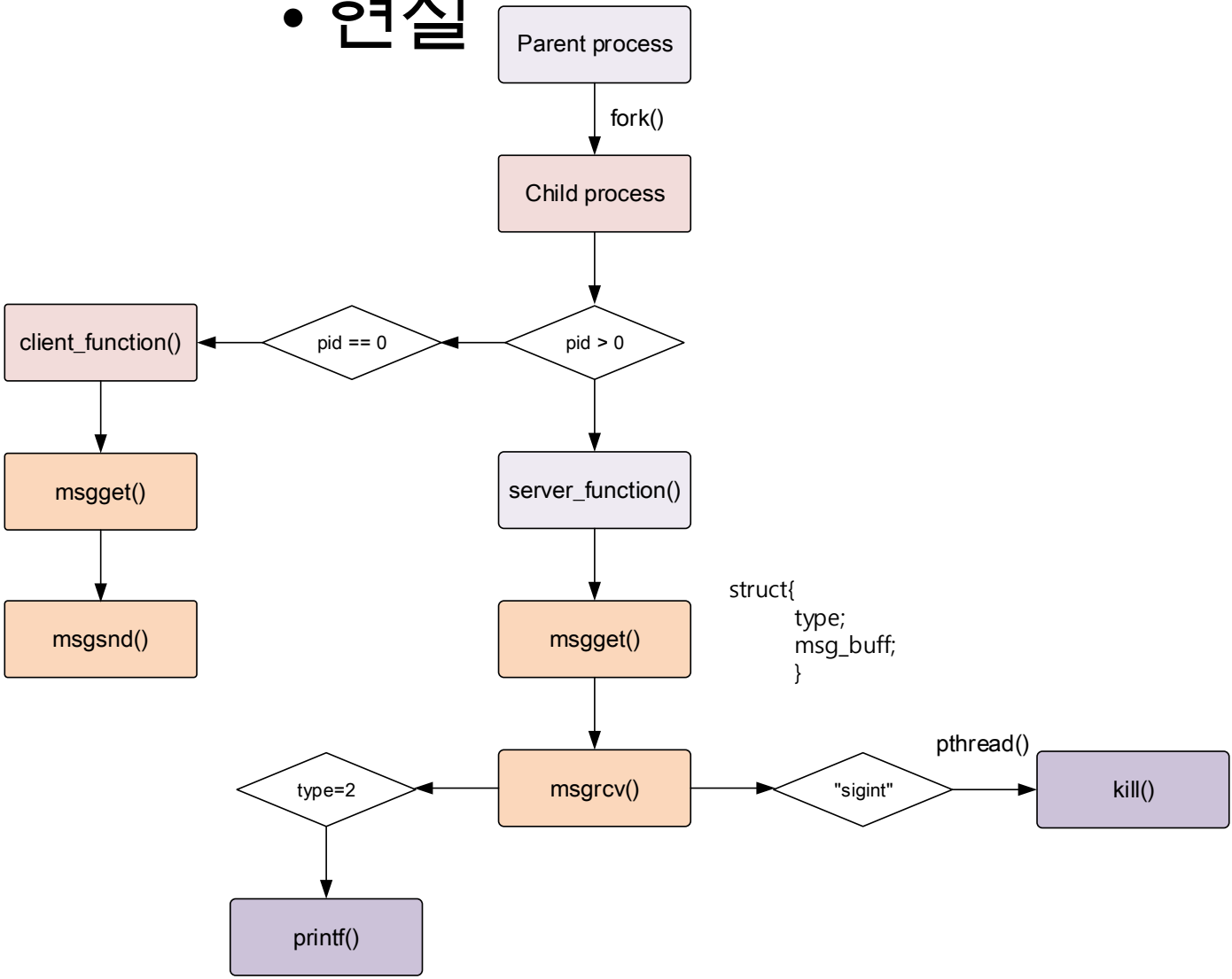


다이어그램

- 꿈…


parent client
createClient()
server
client1
client2
client3
socket()
bind()
listen()
connect()
sock_addr, server_addr, username
Unix Domain Socket Programming
perror() exit()
no
yes
send(clin_info)
"[usrname] is logged in(out)"
mq_open()
pthread_create()
fprinf()
save the log information as a .txt file
fork()
send()
recei ve()
store()
Message Queue
struct mq_attr{ };
SIGINT
fprinf()
save the conversation as a .txt file
sig_handler()

- 현실


Parent process
fork()
Child process
pid > 0
pid == 0
client_function()
server_function()
msgget()
msgsnd()
msgget()
struct{
type;
msg_buff;
}
msgrcv()
type=2
printf()
"sigint"
pthread()
kill()

코드 설명

- 구조체 및 전역변수

```
#define BUFF_SIZE 256

typedef struct{
    long type;
    int msgnum;
    char msg_buff[BUFF_SIZE];
} mq_attr;

int msqid;
```

- 사용함수
  - main( )
  - client_function( )
  - server_function()
  - sighandler()

코드 설명

- int main ( )

```
int main(int argc, char** argv)
{
    int pid = fork();                    // fork하여 자식 프로세스 생성
    int death;

    if(pid > 0)                          // 자식 프로세스가 아니면
    {
        printf("Parents Process: %d, %d\n", getpid(), pid);  //자신pid, 자식 pid
        server_function();               // server_function() 호출
    }
    else if(pid == 0)                    // 자식 프로세스이면
    {
        printf("\nChild Process: %d, %d\n", getpid(), pid);
        client_function();               // client_function() 호출
    }
    else if(pid == -1)
    {
        perror("fork error : ");
        exit(0);
    }

    return 0;
}
```

- 구조체 및 전역변수

코드 설명

- void client ( )

```
void client_function()
{
    mq_attr attr;
    int msqid;

    if(-1 == (msqid = msgget((key_t)0123, IPC_CREAT|0666)))  // 메시지큐 생성
    {
        perror("child mq open fail");
        exit(1);
    }
    else
        printf("child's message queue has opened!\n");
    while(1){
        printf("\nType of the data?: ");
        scanf("%ld", &attr.type);                          // 메시지 타입 입력
        printf("Input Data into the queue: ");
        scanf("%s", attr.msg_buff);                        // 메시지 내용 입력

        if(msgsnd(msqid, &attr, sizeof(struct mq_attr), IPC_NOWAIT) == -1)
        {
            perror("message haven't sent to the queue successfully");
            exit(1);
        }
      sleep(1);      }
}
```

코드 설명

- void server_function ( )

```
void server_function()
{
    mq_attr attr;
    signal(SIGINT, sig_handler);      // SIGINT 입력 시 sig_handler() 호출


    msqid = msgget((key_t)0123, IPC_CREAT|0666);     // 메시지 큐 생성

    if(msqid == -1)
    {
        perror("parent mq open fail");
        exit(0);
    }
    else
        printf("parent's queue has opened!\n");

    while(1) {
        if(msgrcv(msqid, &attr, sizeof(struct mq_attr),2,0) == -1)
       // 메시지 수신 (type 2인 것만)
        {
            perror("couldn't get the messages well\n");
            exit(-1);
        }
        printf("read data from the queue: %s\n", attr.msg_buff); }
}
```

코드 설명

- void sig_handler ( )

```
void sig_handler()
{
  printf("\n\ngot the SIGINT signal, parent process will die");

  msgctl(msqid, IPC_RMID, NULL);    // 메시지 큐 제거

  kill(SIGINT, 0);   // 자식 프로세스 종료

  printf("\nchild process will die, too\n");

  exit(1);           // 부모 프로세스 종료
}
```

감사합니다!