오픈소스리눅스프로그래밍 기말 과제

- IPC를 활용한 프로그래밍-

최서윤 (seoyun@pel.smuc.ac.kr)

상명대학교 컴퓨터공학과 201621173

목 차

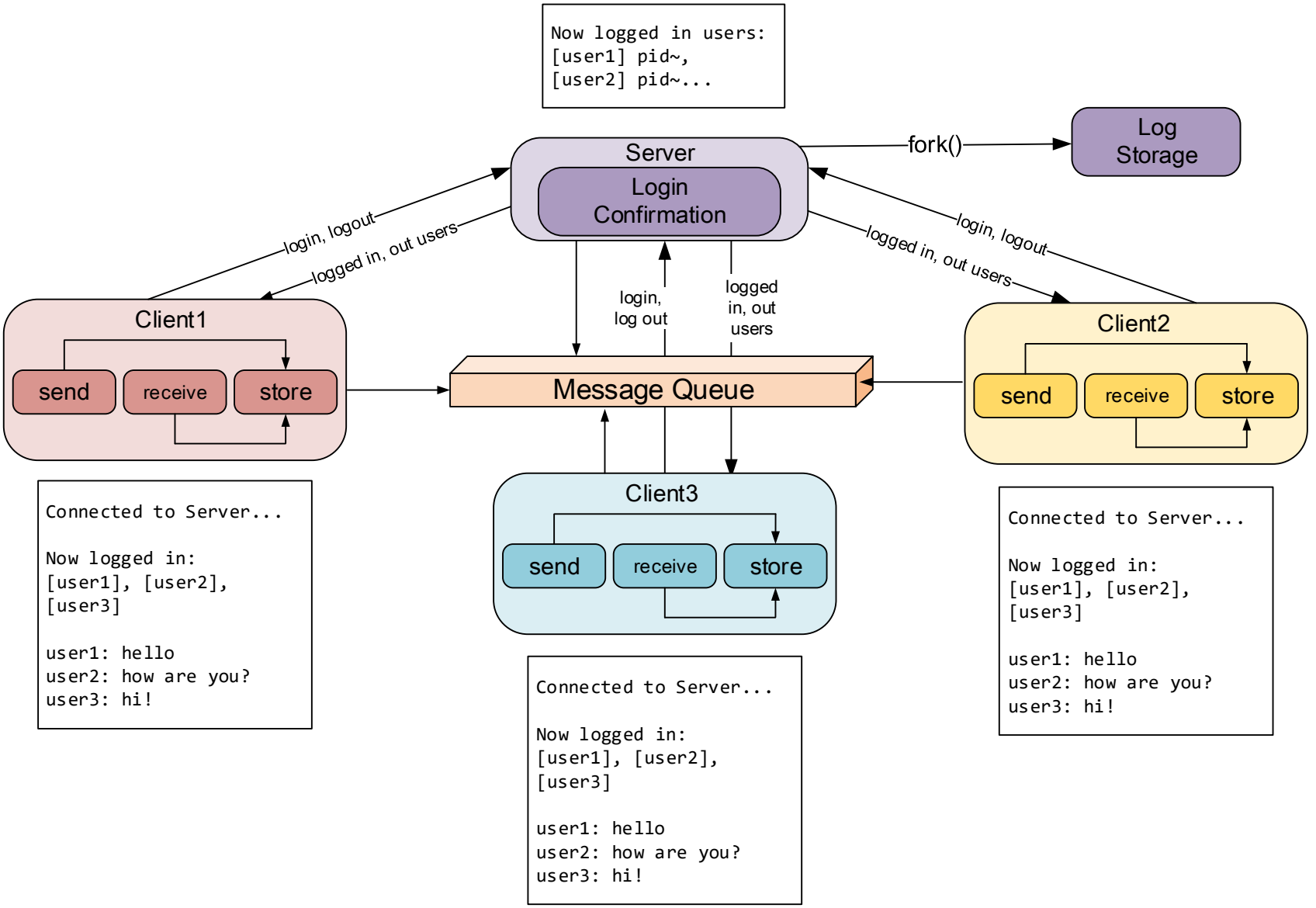
- 개요
- 아키텍처
- 다이어그램
- 세부 구현 계획

개요

- 주제
  - 메시지 큐 기반의 단체 채팅 프로그램
  - 서버-클라이언트 구조

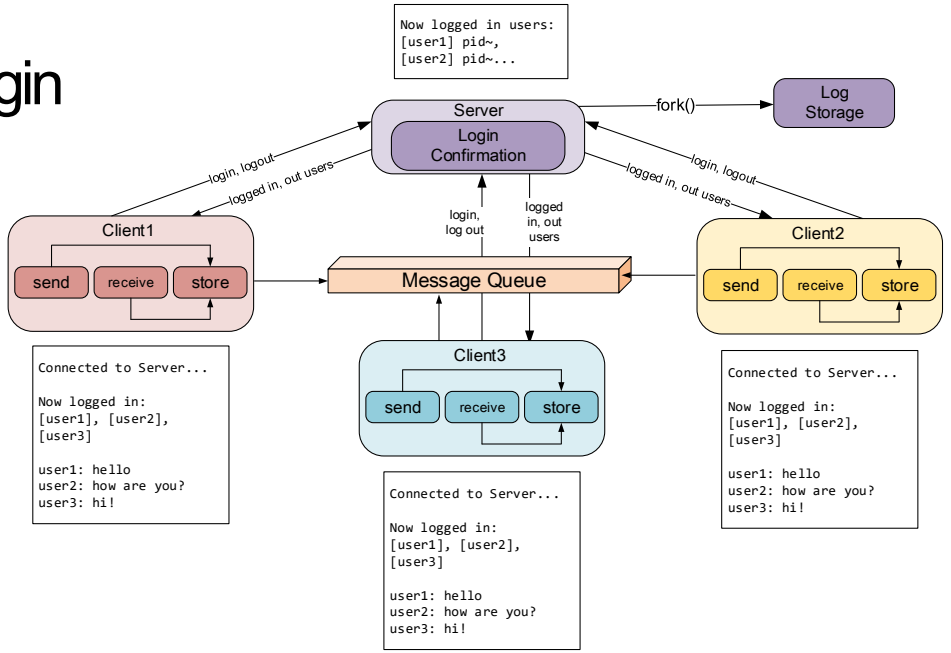
- 기술 설정
  - IPC 관련 기술
    - PIPE, FIFO, 공유메모리, 메시지큐, 시그널, 세마포어
  - 기타 기술 활용
    - fork(), pthread(), mutex, Unix Domain Socket…

아키텍처

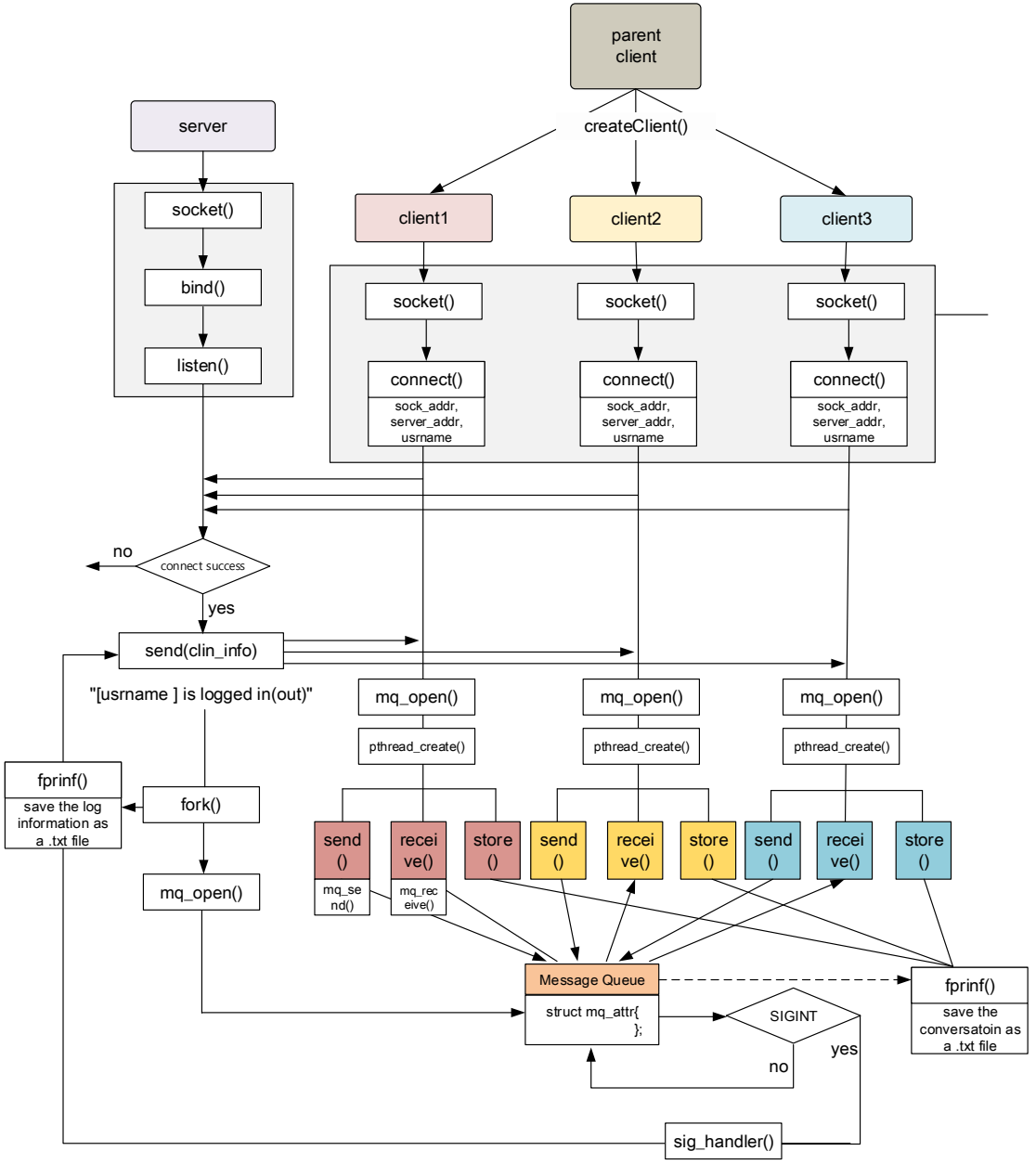

Now logged in users:
[user1] pid~,
[user2] pid~...
Server
Login Confirmation
fork()
Log Storage
login, logout
logged in, out users
login, logout
logged in, out users
login, log out
logged in, out users
Client1
send
receive
store
Message Queue
Client2
send
receive
store
Client3
send
receive
store
Connected to Server...
Now logged in:
[user1], [user2],
[user3]
user1: hello
user2: how are you?
user3: hi!
Connected to Server...
Now logged in:
[user1], [user2],
[user3]
user1: hello
user2: how are you?
user3: hi!
Connected to Server...
Now logged in:
[user1], [user2],
[user3]
user1: hello
user2: how are you?
user3: hi!

아키텍처

- 클라이언트, 서버 소켓 연결
  - 클라이언트 프로세스 포크 시 서버에 login
  - 서버는 login 및 logout 유저 알림
- 메시지 큐 생성
  - 클라이언트간 메시지 큐를 통해 채팅
  - 서버에서 채팅방에 단방향 통보 가능
- 스레드를 통한 분할 작업
  - send, receive, store 스레드를 이용하여 작업 분할 처리
- 로그 정보 저장
  - 서버 프로세스의 로그 기록 txt 파일로 출력



다이어그램


parent client
createClient()
server
socket()
bind()
listen()
client1
client2
client3
socket()
socket()
socket()
connect()
sock_addr, server_addr, usrname
connect()
sock_addr, server_addr, usrname
connect()
sock_addr, server_addr, usrname
no
connect success
yes
send(clin_info)
"[usrname] is logged in(out)"
mq_open()
mq_open()
mq_open()
pthread_create()
pthread_create()
pthread_create()
fprinf()
save the log information as a .txt file
fork()
mq_open()
send()
mq_send()
receive()
mq_receive()
store()
send()
receive()
store()
send()
receive()
store()
Message Queue
struct mq_attr{ };
SIGINT
yes
no
fprinf()
save the conversatoin as a .txt file
sig_handler()

세부 구현 계획

- Unix Domain Socket 프로그래밍을 통해 서버, 클라이언트 연결
  - 서버 소켓 프로세스
    - socket(), bind(), listen()
  - 클라이언트 소켓 프로세스
    - socket(), connect()
    - 클라이언트 프로세스 fork() 시 서버 프로세스에 login (struct 내에 소켓, 서버 정보 및 사용할 유저 이름 추가하여 서버로 socket 전송)
  - 소켓 연결 이후 서버에서 login 된 클라이언트 정보 send()하여 모든 클라이언트에게 알림
- 메시지 큐 생성
  - 클라이언트 프로세스에서 mq_create()를 통해 클라이언트들 간 채팅을 위한 메시지 큐 생성
    - 서버에서도 동일 메시지 큐를 통해 단방향 알림 가능
    - mq_send(), mq_receive()를 통해 메시지 큐로부터 메시지 송수신 가능

세부 구현 계획

- 스레드를 통한 분할작업
  - pthread_create()를 통하여 send, receive, store 스레드 생성
    - send 스레드에서 mq_send()를 통해 메시지 송신
    - receive 스레드에서 mq_receive를 통해 메시지 수신
    - store 스레드에서 fscanf()를 통해 모든 대화 내용 파일 출력
  - mutex를 통해 스레드 synchronize
- 로그 정보 저장
  - 서버 프로세스를 fork()시켜 로그 정보 저장 역할을 하는 자식 프로세스 생성
    - semaphore를 통해 프로세스 synchronize
    - 부모 프로세스로부터 클라이언트가 login했을 때의 시각을 struct 형태로 전달 받음
    - 클라이언트로부터 logout했을 때의 시각(SIGINT 입력 시각), 클라이언트의 유저 이름 등을 구조체로 전달 받음
    - fscanf()를 통해 클라이언트의 유저 이름, login 시각, logout 시각 정보 파일 출력
  - 클라이언트가 SIGINT 시그널 입력 시 해당 클라이언트 logout
    - 서버 프로세스는 logout한 유저를 모든 클라이언트에게 알림
    - sig_handler()를 통해 서버 프로세스에서 fork()된 자식프로세스에 유저의 SIGINT 입력 시각 전달

감사합니다!