



REPORT

운영체제 과제 #2

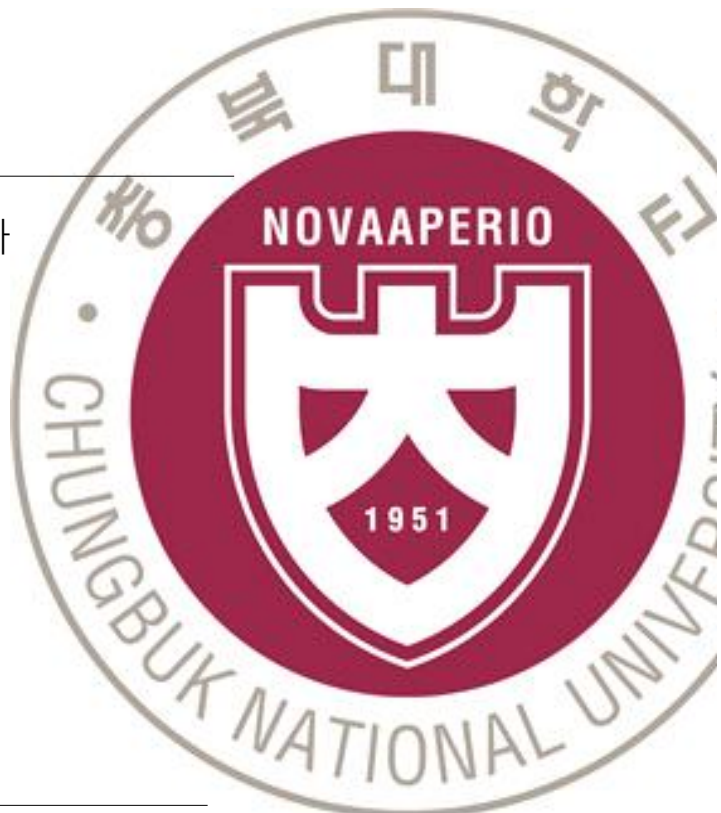
제출일 2022.03.23

학과 소프트웨어학과

학번 2020039009

이름 차현아

담당교수 이건명 교수님



3장 강의노트의 모든 프로그램 실행해보기

1. UNIX/LINUX에서 새로운 프로세스를 fork하는 C프로그램

코드

```
chahyeona@chahyeona-VirtualBox: ~/os_homework
#include<sys/wait.h>
#include<stdio.h>
#include<unistd.h>
#include<stdlib.h>
int main()
{
    int pid;
    int status;

    pid=fork();
    if(pid<0)
    {
        fprintf(stderr, "Fork Failed");
        exit(-1);
    }
    else if(pid==0)
    {
        printf("Child : %d Parent : %d \n", getpid(), getppid());
        exit(3);
    }
    else
    {
        wait(&status);
        printf("Parent: %d Child: %d\n", getpid(), pid);
        printf("Status : %d\n", status>>8);
    }
    return 0;
}
```

실행화면

```
chahyeona@chahyeona-VirtualBox: ~/os_homework
chahyeona@chahyeona-VirtualBox:~/os_homework$ ./a.out
Child : 6994 Parent : 6993
Parent: 6993 Child: 6994
Status : 3
chahyeona@chahyeona-VirtualBox:~/os_homework$
```

2. 프로세스를 fork한 다음, 새로운 프로그램을 실행하는 C프로그램

코드

```
chahyeona@chahyeona-VirtualBox: ~/os_homework
#include<stdio.h>
#include<stdlib.h>
#include<sys/wait.h>
#include<unistd.h>

int main()
{
    int pid;
    pid=fork();
    if(pid<0)
    {
        fprintf(stderr, "Fork Failed");
        exit(-1);
    }
    else if(pid==0)
    {
        execlp("/bin/ls", "ls", NULL);
    }
    else
    {
        wait(NULL);
        printf("Child Complete\n");
        exit(0);
    }
}
```

실행화면

```
chahyeona@chahyeona-VirtualBox: ~/os_homework
chahyeona@chahyeona-VirtualBox:~/os_homework$ vi execlp.c
chahyeona@chahyeona-VirtualBox:~/os_homework$ ls
a.out  exec.c  execlp.c  fork1.c  signal.c
chahyeona@chahyeona-VirtualBox:~/os_homework$ ./a.out
a.out  exec.c  execlp.c  fork1.c  signal.c
Child Complete
chahyeona@chahyeona-VirtualBox:~/os_homework$
```

3. 시그널

코드

```
chahyeona@chahyeona-VirtualBox: ~/os_homework

#include<stdlib.h>
#include<stdio.h>
#include<unistd.h>
#include<signal.h>

void signal_handler(int signum)
{
    printf("Caught signal %d\n", signum);
    exit(signum);
}

int main()
{
    signal(SIGINT, signal_handler);

    while(1)
    {
        printf("Program processing here.\n");
        sleep(1);
    }
    return 0;
}
```

실행화면

```
chahyeona@chahyeona-VirtualBox: ~/os_homework

chahyeona@chahyeona-VirtualBox:~/os_homework$ ./a.out
Program processing here.
Program processing here.
Program processing here.
Program processing here.
Program processing here.
Program processing here.
^CCaught signal 2
chahyeona@chahyeona-VirtualBox:~/os_homework$
```

4. 파이프

코드

```
chahyeona@chahyeona-VirtualBox: ~/os_homework
#include<stdio.h>
#include<string.h>
#include<unistd.h>

#define BUFFER_SIZE 25
#define READ_END 0
#define WRITE_END 1

int main()
{
    char write_msg[BUFFER_SIZE]="Greetings";
    char read_msg[BUFFER_SIZE];
    int fd[2];
    int pid;

    if(pipe(fd)==-1)
    {
        fprintf(stderr, "Pipe Failed\n");
        return 1;
    }

    pid=fork();

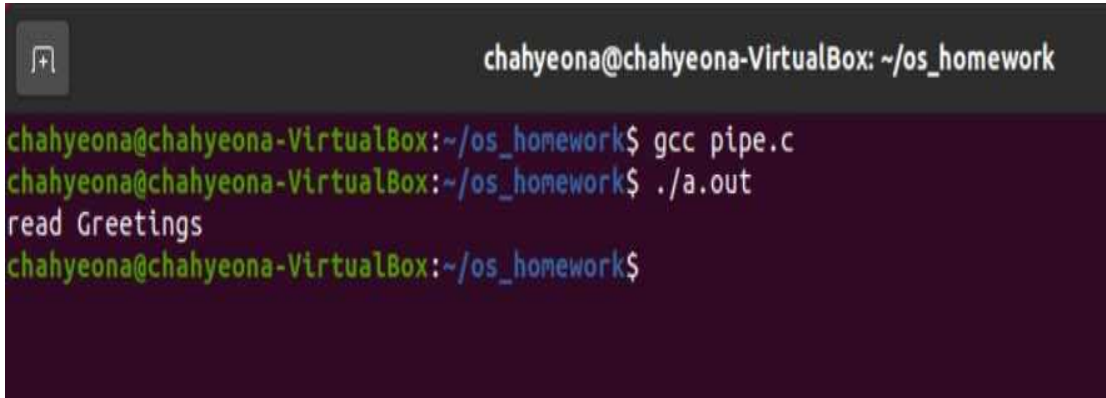
    if(pid<0)
    {
        fprintf(stderr, "Fork Failed \n");
        return 1;
    }
}
```

```
chahyeona@chahyeona-VirtualBox: ~/os_homework
{
    fprintf(stderr, "Pipe Failed\n");
    return 1;
}

pid=fork();

if(pid<0)
{
    fprintf(stderr, "Fork Failed \n");
    return 1;
}
else if(pid==0)
{
    close(fd[READ_END]);
    write(fd[WRITE_END], write_msg, strlen(write_msg)+1);
    close(fd[WRITE_END]);
}
else
{
    close(fd[WRITE_END]);
    read(fd[READ_END], read_msg, BUFFER_SIZE);
    printf("read %s\n", read_msg);
    close(fd[READ_END]);
}
return 0;
}
```

실행화면



```
chahyeona@chahyeona-VirtualBox: ~/os_homework
chahyeona@chahyeona-VirtualBox:~/os_homework$ gcc pipe.c
chahyeona@chahyeona-VirtualBox:~/os_homework$ ./a.out
read Greetings
chahyeona@chahyeona-VirtualBox:~/os_homework$
```

5. 소켓 : 가시성을 위해 기존 코드에서 출력을 추가하였습니다.

: 컴퓨터 한대로 통신을 하기 위해서 getLocalHost()를 사용하여 제 컴퓨터 IP 주소를 불러왔습니다.

코드 - DateClient 클래스

```
import java.net.*;
import java.io.*;
public class DateClient {
    public static void main(String[] args) {
        BufferedReader bin2 = null; // 키보드로부터 읽어들이기 위한 입력스트림
        PrintWriter out = null; // 서버로 내보내기 위한 출력 스트림
        BufferedReader bin=null;
        Socket sock=null;
        try {
            sock = new Socket(InetAddress.getLocalHost(), 6013);
            InputStream in = sock.getInputStream();
            bin = new BufferedReader(new InputStreamReader(in));
            bin2 = new BufferedReader(new InputStreamReader(System.in));
            out = new PrintWriter(new BufferedWriter(new
            OutputStreamWriter(sock.getOutputStream())));

            System.out.print("서버로 보낼 메시지: ");
            String data = bin2.readLine(); // 키보드로부터 입력
            out.println(data);
            out.flush();

            String str2 = bin.readLine(); // 서버로부터 되돌아오는 데이터를 읽어들이
            System.out.println("서버로부터 되돌아온 메시지: " + str2);
            sock.close();
        }
        catch (IOException ioe) {
            System.err.println(ioe);
        }
    }
}
```

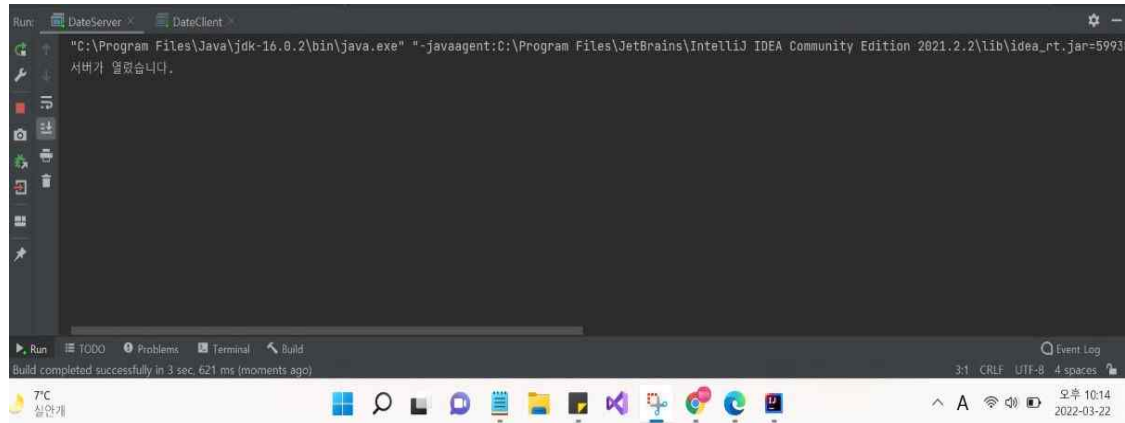
```
}  
}
```

코드 - DateServer 클래스

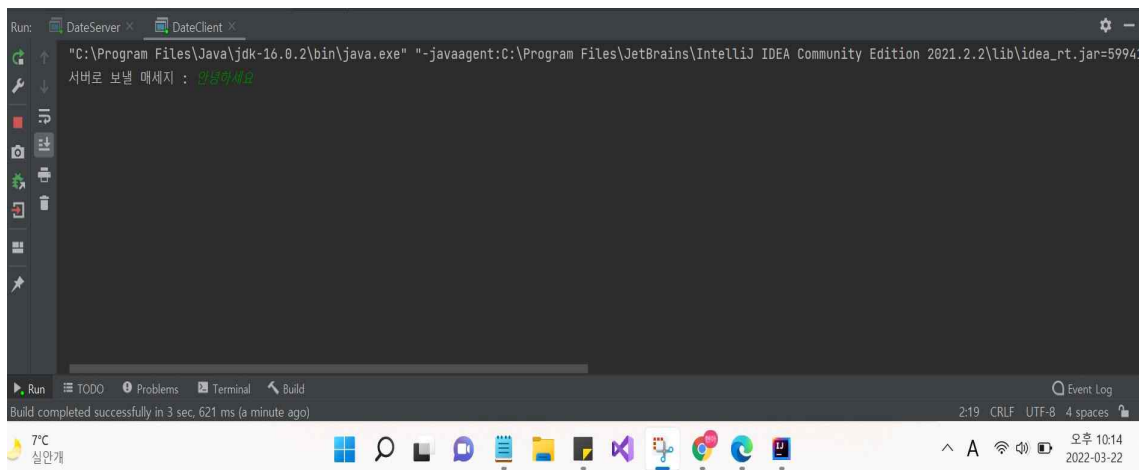
```
import java.net.*;  
import java.io.*;  
public class DateServer {  
    public static void main(String[] args) {  
        ServerSocket sock=null;  
        Socket socket = null;  
        BufferedReader in = null;  
        PrintWriter out = null;  
        try {  
            sock = new ServerSocket(6013);  
  
            System.out.println("서버가 열렸습니다.");  
            socket = sock.accept(); // 서버를 생성, Client는 접속 대기  
  
            in = new BufferedReader(new InputStreamReader(socket.getInputStream()));  
  
            out = new PrintWriter(new BufferedWriter(new  
                OutputStreamWriter(socket.getOutputStream())));  
  
            String str = null;  
            str = in.readLine();  
  
            System.out.println("Client로부터 온 메시지: " + str);  
  
            out.write(str);  
            out.flush();  
            socket.close();  
        }  
        catch (IOException ioe) {  
            System.err.println(ioe);  
            System.out.println("해당 포트가 열려있습니다!");  
        }  
    }  
}
```

실행화면

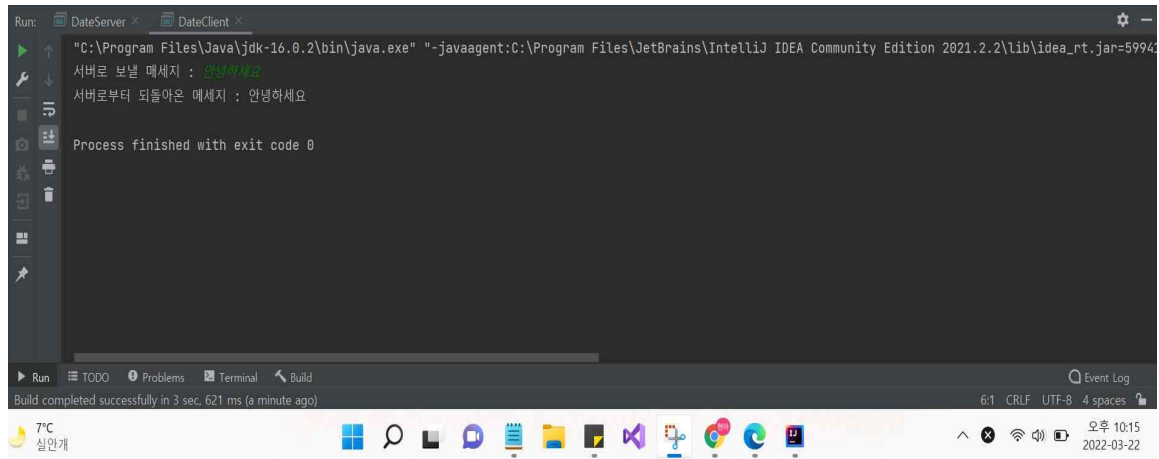
1. 서버 열기



2. DateClient 실행 -> 서버로 보낼 메시지 입력

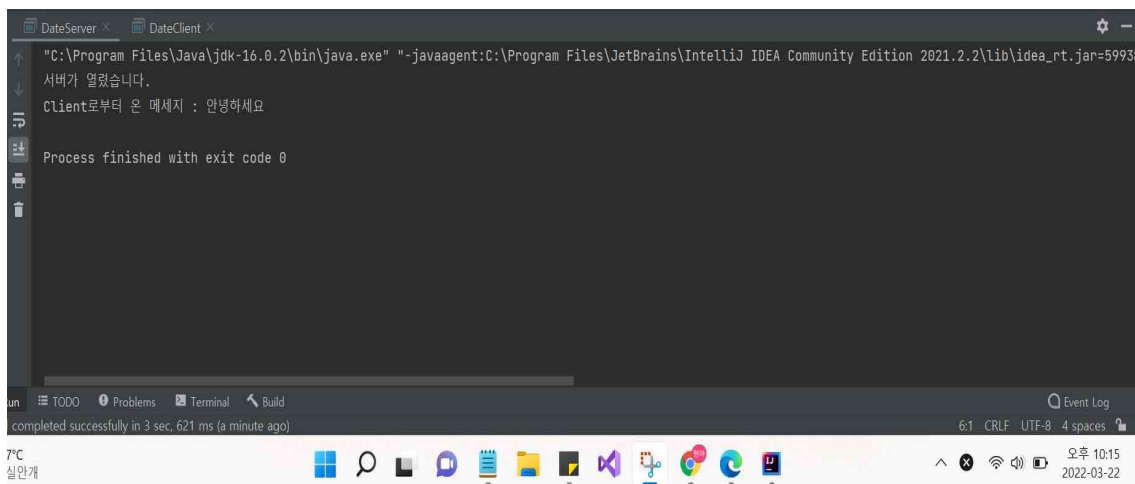


3. 서버에게 되돌아온 메시지



```
Run: DateServer x DateClient x
"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.2.2\lib\idea_rt.jar=59940:C:\Program Files\Java\jdk-16.0.2\bin" -Dfile.encoding=UTF-8
서버로 보낸 메시지 : 20220322
서버로부터 되돌아온 메시지 : 안녕하세요
Process finished with exit code 0
Build completed successfully in 3 sec, 621 ms (a minute ago)
```

4. 서버가 클라이언트에게 메시지를 받음



```
Run: DateServer x DateClient x
"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.2.2\lib\idea_rt.jar=59938:C:\Program Files\Java\jdk-16.0.2\bin" -Dfile.encoding=UTF-8
서버가 열렸습니다.
Client로부터 온 메시지 : 안녕하세요
Process finished with exit code 0
Build completed successfully in 3 sec, 621 ms (a minute ago)
```

6. JAVA RMI

코드 - RmiClient 클래스

```
import java.rmi.Naming;
public class RmiClient { // RMI Client
public static void main(String args[]) throws Exception {
RmiServerIntf obj = (RmiServerIntf)
Naming.lookup("//localhost/RmiServer");
System.out.println(obj.getMessage());
}
}
```

코드 - RmiServer 클래스

```
import java.rmi.Naming;
import java.rmi.RemoteException;
import java.rmi.server.UnicastRemoteObject;
import java.rmi.registry.*;
public class RmiServer extends UnicastRemoteObject implements RmiServerIntf {
public static final String MESSAGE = "Hello World";
public RmiServer( ) throws RemoteException {
super(0);
}
public String getMessage( ) {
return MESSAGE;
}
public static void main(String args[]) throws Exception {
System.out.println("RMI server started");
try {
LocateRegistry.createRegistry(1099);
System.out.println("java RMI registry created.");
} catch (RemoteException e) {
System.out.println("java RMI registry already exists.");
}

RmiServer obj = new RmiServer();
Naming.rebind("//localhost/RmiServer", obj);
System.out.println("PeerServer bound in registry");
}
}
```

코드 – RmiServerIntf 인터페이스

```
import java.rmi.Remote;  
import java.rmi.RemoteException;  
public interface RmiServerIntf extends Remote {  
    public String getMessage() throws RemoteException;  
}
```

실행화면

