

설계과제 2 개요 : Soongsil Converter

Linux System Programming, School of CSE, Soongsil University, Spring 2019

○ 개요

- Java 언어 프로그램을 C 언어 프로그램으로 자동 변환

○ 목표

- 유닉스/리눅스 시스템에서 제공하는 여러 함수를 사용하여 텍스트 파일에서 읽어 들인 문자열을 비교하고 그에 상응하는 문자열을 별도의 파일에 작성함으로써 시스템 프로그래밍 설계 및 응용 능력을 향상

○ 팀 구성

- 개인별 프로젝트

○ 개발환경

- OS : Ubuntu 16.04
- Tools : vi(m), gcc, gdb

○ 보고서 제출 방법

- 설계과제는 "보고서.hwp"(개요, 상세설계, 구현방법, 결과 및 소스코드와 실행결과가 함께 있는 워드(hwp 또는 MS-Word) 파일)와 "소스코드" (makefile, obj, *.c, *.h 등 컴파일하고 실행하기 위한 모든 파일)를 제출해야 함
- 모든 설계과제 결과물은 "#P설계과제번호_학번_버전.zip"(예. #P2_20160000_v1.0.zip)형태로 파일 이름을 명명하고, zip프로그램으로 압축하여 제출해야 함.
- 압축파일 내 "보고서" 디렉토리 and "소스코드" 디렉토리 2개 만들어 제출해야 함
- 제출한 압축 파일을 풀었을 때 해당 디렉토리에서 컴파일 및 실행이 되어야 함. 해당 디렉토리에서 컴파일이나 실행되지 않을 경우, 기본과제 및 설계과제 제출 방법을 따르지 않는 경우 감점 20% 외 추가 20% 감점
- 기타 내용은 syllabus 참고

○ 제출 기한

- 5월 1일(월) 오후 11시 59분 59초 (서버 시간이 30분 정도 빠를 수 있기 때문에 1시간 지연 허용)

○ 보고서 양식

- 보고서는 다음과 같은 양식으로 작성

- | |
|---|
| <ol style="list-style-type: none">1. 과제 개요 // 위 개요를 더 상세하게 작성2. 설계 // 함수 기능별 흐름도(순서도) 반드시 포함3. 구현 // 함수 프로토타입 반드시 포함4. 테스트 및 결과 // 테스트 프로그램의 실행 결과 캡처 및 분석5. 소스코드 // 주석 |
|---|

○ Ubuntu 16.04에서 Java 설치 및 실행

- openjdk7 설치

```
$ sudo add-apt-repository ppa:openjdk-r/ppa
$ sudo apt-get update
$ sudo apt-get install openjdk-7-jdk
$ java -version
```

실행 예시 <java 버전 확인>

```
oslab@oslab-localhost:~/2019/lsp_2$ java -version
java version "1.7.0_95"
OpenJDK Runtime Environment (IcedTea 2.6.4) (7u95-2.6.4-3)
OpenJDK 64-Bit Server VM (build 24.95-b01, mixed mode)
```

- Java 환경변수 지정

```
$ sudo vi /etc/environment
JAVA_HOME = "/usr/lib/jvm/java-7-openjdk-amd64"
**** Path 확인 후 위와 같이 environment에 추가 ****
$ source /etc/environment
$ echo $JAVA_HOME
```

실행 예시 <java 환경 변수 지정 확인>

```
oslab@oslab-localhost:~/2019/lsp_2$ echo $JAVA_HOME
/usr/lib/jvm/java-7-openjdk-amd64
oslab@oslab-localhost:~/2019/lsp_2$
```

- Java 소스 파일 생성 및 실행

```
SSU_example.java
class SSU_example{
    public static void main(String[] args){
        System.out.printf("Hello, Java in Ubuntu\n");
    }
}
```

실행 예시 <java 소스 생성 및 컴파일 과정>

```
oslab@oslab-localhost:~/2019/lsp_2$ vi SSU_example.java
oslab@oslab-localhost:~/2019/lsp_2$ javac SSU_example.java
oslab@oslab-localhost:~/2019/lsp_2$ ls
SSU_example.class  SSU_example.java
oslab@oslab-localhost:~/2019/lsp_2$ java SSU_example
Hello, Java in Ubuntu
oslab@oslab-localhost:~/2019/lsp_2$
```

○ 설계 및 구현

- ssu_convert <FILENAME> [OPTION]

1) <FILENAME> : 변환할 JAVA 언어로 작성된 프로그램 파일명. 절대경로와 상대경로 모두 입력 가능해야함. 존재하지 않는 파일을 대상으로 할 경우 예외처리

- ssu_convert는 다음과 같은 Java 프로그램 파일을 C 언어 프로그램으로 변환시킬 수 있어야 함

- ✓ q1.java - 짝수 및 홀수 합 구하는 프로그램
 - 표준 입출력 함수

- If-else(if)-else 문
- for 문
- ✓ q2.java - Stack 구현 프로그램
 - q1.java에서 사용된 기능 포함
 - 함수 호출
 - 전처리기
- ✓ q3.java - *.txt 파일 생성 프로그램
 - q1.java, q2.java에서 사용된 기능 포함
 - 파일 입출력
 - java 프로그래밍 파일에서 IOException은 C 언어 프로그램에서 파일이 존재하지 않는 예외인 경우만 처리
- header 자동 생성 (개발자가 미리 만든 참조 헤더 테이블 이용)

예시 <변환 시 참조되는 헤더 테이블>
<pre>open #include <sys/types.h> #include <sys/stat.h> #include <fcntl.h> read #include <unistd.h> exit #include <stdlib.h></pre>

- ✓ ssu_convert는 Java 언어 프로그램을 C 언어 프로그램으로 변환 시 개발자가 만든(헤더 테이블은 자동으로 생성하지 않고 별도로 참조되는 테이블로 만들어야 함) 헤더 테이블을 참조하며 프로그램에서 필요한 헤더는 자동으로 생성해야 함
- ✓ header는 위 형식을 그대로 지키되 개발자가 필요한 함수들과 헤더를 자유롭게 추가 및 삭제 가능
- makefile 자동 생성

변환할 java 프로그램 파일		변환된 C 프로그램 파일
q1.java	public class q1	q1.c
q2.java	class Stack	Stack.c
	public class q2	q2.c
q3.java	public class q3	q3.c

실행 예시 <ssu_convert 수행 결과 파일>
<pre>oslab@oslab-localhost:~/2019/lsp_2\$ ls Makefile header q1.java q2.java q3.java q3java.txt ssu_convert.c ssu_convert.h oslab@oslab-localhost:~/2019/lsp_2\$ make gcc -c -o ssu_convert.o ssu_convert.c gcc -o ssu_convert ssu_convert.o oslab@oslab-localhost:~/2019/lsp_2\$ ls Makefile header q1.java q2.java q3.java q3java.txt ssu_convert ssu_convert.c ssu_convert.h ssu_convert.o oslab@oslab-localhost:~/2019/lsp_2\$./ssu_convert q1.java oslab@oslab-localhost:~/2019/lsp_2\$ ls Makefile header q1.c q1.java q1.Makefile q2.java q3.java q3java.txt ssu_convert ssu_convert.c ssu_convert.h ssu_convert.o oslab@oslab-localhost:~/2019/lsp_2\$</pre>

- ✓ 변환된 C 언어 프로그램 파일을 컴파일 할 수 있는 '파일명_Makefile'도 함께 생성되어야 함
- ✓ java 언어 프로그램에서 사용되는 표준출력함수는 'System.out.printf("format-string", arg1, arg2, ...);' 만 허용하며 표준출력함수는 C 언어 프로그램에서 printf로 변경되어야 함
- ✓ Java 언어 프로그램에서 사용되는 표준 입력 함수는 'Scanner 변수이름 = new

Scanner(System.in)' '변수이름.nextInt'만 허용하며 표준출력함수는 C 언어 프로그램에서 scanf로 변경되어야 함

- ✓ Java 언어 프로그램에서 사용되는 syntax가 C 언어 프로그램에서 사용되지 않으면 무시

예시 <C 언어에서 제공되지 않는 Java 프로그램 코드>

```
import java.util.Scanner;
Scanner scn = new Scanner(System.in);
```

- 옵션 [OPTION]

- ✓ 옵션이 없는 경우

실행 예. 옵션 없는 경우

```
oslab@oslab-localhost:~/2019/lsp_2$ ./ssu_convert q1.java
q1.c convert Success!
oslab@oslab-localhost:~/2019/lsp_2$
oslab@oslab-localhost:~/2019/lsp_2$
```

- ✓ -j 옵션 : 변환할 Java 언어 프로그램 코드 출력

실행 예. -j 옵션

```
oslab@oslab-localhost:~/2019/lsp_2$ ./ssu_convert q1.java -j
q1.c convert Success!
1 import java.util.Scanner;
2
3 public class q1{
4     public static void main(String[] args){
5         Scanner scn = new Scanner(System.in);
6
7         System.out.printf("Enter the number : ");
8
9         int num;
10        num = scn.nextInt();
11        int even=0, odd=0;
12
13        for(int i=1; i<=num; i++){ // Checking...
14            if(i % 2 == 0){
15                even+=i;
16            }
17            else{
18                odd+=i;
19            }
20        }
21
22        System.out.printf("Sum of Even number : %d\n", even);
23        System.out.printf("Sum of Odd number : %d\n", odd);
24
25        return ;
26    }
27 }
```

✓ -c 옵션 : 변환된 C 언어 프로그램 출력

실행 예. -c 옵션

```
oslab@oslab-localhost:~/2019/lsp_2$ ./ssu_convert q1.java -j
q1.c convert Success!
1 import java.util.Scanner;
2
3 public class q1{
4     public static void main(String[] args){
5         Scanner scn = new Scanner(System.in);
6
7         System.out.printf("Enter the number : ");
8
9         int num;
10        num = scn.nextInt();
11        int even=0, odd=0;
12
13        for(int i=1; i<=num; i++){ // Checking...
14            if(i % 2 == 0){
15                even+=i;
16            }
17            else{
18                odd+=i;
19            }
20        }
21
22        System.out.printf("Sum of Even number : %d\n", even);
23        System.out.printf("Sum of Odd number : %d\n", odd);
24
25        return ;
26    }
27 }
oslab@oslab-localhost:~/2019/lsp_2$
```

✓ -p : Java 언어 프로그램에서 사용된 함수들을 C 언어 프로그램에서 대응되는 함수와 함께 출력

실행 예. -p 옵션

```
oslab@oslab-localhost:~/2019/lsp_2$ ./ssu_convert q1.java -p
q1.c convert Success!
1 System.out.printf() -> printf()
2 scn.nextInt() -> scanf()
oslab@oslab-localhost:~/2019/lsp_2$
```

✓ -f 옵션 : Java 언어 프로그램 파일 및 C 언어 프로그램 파일의 파일 크기 출력

실행 예. -f 옵션

```
oslab@oslab-localhost:~/2019/lsp_2$ ./ssu_convert q1.java -f
q1.c convert Success!
q1.java file size is 466 bytes
q1.c file size is 350 bytes
oslab@oslab-localhost:~/2019/lsp_2$
oslab@oslab-localhost:~/2019/lsp_2$
```

- ✓ -l 옵션 : Java 언어 프로그램 파일 및 C 언어 프로그램 파일의 라인 수 출력

실행 예. -l 옵션

```
oslab@oslab-localhost:~/2019/lsp_2$ ./ssu_convert q1.java -l
q1.c convert Success!
q1.java line number is 27 lines
q1.c line number is 25 lines
oslab@oslab-localhost:~/2019/lsp_2$
oslab@oslab-localhost:~/2019/lsp_2$
```

- ✓ -r 옵션 : 자식 프로세스를 생성하여 Java 언어 프로그램이 변환되는 과정을 한 줄 한줄 출력. 단, Java 언어 프로그램의 매 statement 가 C 언어 프로그램으로 변환되어 출력되고 터미널 화면이 clear 되고 다시 다음 statement 변환 및 출력. 단, clear는 system 함수 사용 가능
- ✓ 변환이 완료되면 '파일명.c converting is finished!' 출력

실행 예. -r 옵션

```
q1.java Converting....
-----
q1.java
-----
1 import java.util.Scanner;
2
3 public class q1{
4     public static void main(String[] args){
5         Scanner scn = new Scanner(System.in);
6
7         System.out.printf("Enter the number : ");
8     }
9 }
-----
q1.c
-----
1 #include <stdio.h>
2 int main(void){
3
4     printf("Enter the number : ")
```

- 기타

- ✓ 모든 옵션은 동시에 사용 가능해야 함. 단 여러 개의 옵션을 함께 사용할 때는 -r 옵션이 가장 먼저 출력되어야 함

실행 예. 여러 옵션을 동시에 사용할 경우

```
oslab@oslab-localhost:~/2019/lsp_2$ ./ssu_convert q1.java -f -l
q1.c convert Success!
q1.java file size is 466 bytes
q1.c file size is 350 bytes
q1.java line number is 27 lines
q1.c line number is 25 lines
oslab@oslab-localhost:~/2019/lsp_2$
oslab@oslab-localhost:~/2019/lsp_2$
```

- ✓ 존재하지 않는 파일 또는 옵션 사용 시 에러 처리
- ✓ clear를 제외하고 system()함수 사용 금지
- ✓ 게시판에 통해 변환되어야 하는 Java 언어 프로그램 이용

<참고> 과제 구현에 필요한 함수(필수 사용 아님)

1. fork() : 리눅스 시스템에서 새로운 프로세스를 생성할 때 사용하는 함수 p.233

```
#include <unistd.h>
pid_t fork(void);
```

리턴값: 자식 프로세스의 경우 0, 부모 프로세스의 경우 자식의 프로세스 ID,
오류 시 -1을 리턴하고 errno가 설정됨

fork_getpid_example.c

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
int main(void)
{
    int pid;
    pid = fork();
    if (pid > 0) {
        printf("Parent : %d -> fork() -> : %d\n", getpid(), pid);
        sleep(1);
    }
    else if (pid == 0)
        printf("Child : %d\n", getpid());
    else if (pid == -1) {
        perror("fork error : ");
        exit(0);
    }
    exit(0);
}
```

실행결과

```
oslab@oslab-localhost:~$ ./fork_getpid_example
Parent : 10220 -> fork() -> 10221
Child : 10221
```

2. gettimeofday() : p.519

3. strstr() : 문자열에서 부분 문자열의 위치를 찾는 함수

```
#include <string.h>
char *strstr(const char * str1, char const * str2);
```

리턴값: 찾고자 하는 문자열이 발견된 첫번째 위치의 포인터를 반환합니다. 찾지 못하면 NULL을 반환

strstr_example.c

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
```

```
int main( void)
{
```

<pre>char *str = "Linux System Programming"; printf("%s\n", strstr(str, "Linux")); exit(0); }</pre>
실행결과
<pre>oslab@oslab-localhost:~\$./strstr_example Linux System Programming</pre>

4. system() : p.270

<pre>#include <stdlib.h> int system(const char* string);</pre>	리턴값: 성공시 0이 아닌 값, 실패시 0
--	-------------------------

5. fopen() : 파일을 오픈하여 파일과 관련된 스트림을 생성하고 버퍼를 할당 p.160

<pre>#include <stdio.h> FILE *fopen(const char * pathname, const char * mode);</pre>	리턴값: 성공 시 파일 포인터, 에러 시 NULL을 리턴하고 errno가 설정됨
--	--

ssu_test.txt
Linux System Programming!
Unix System Programming!
Linux Mania
Unix Mania
fopen_example.c
<pre>#include <stdio.h> #include <stdlib.h> int..main(void) { char* fname = "ssu_test.txt"; char *mode = "r"; if(fopen(fname, mode) == NULL){ fprintf(stderr, "fopen error for %s\n", fname); exit(1); } else printf("Sucess!\nFilename: <%s>, mode <%s>\n", fname, mode); exit(0); }</pre>
실행결과
<pre>oslab@oslab-localhost:~\$./fopen_example Success! Filename: <ssu_test.txt>, mode: <r></pre>

○ 보고서 제출 시 유의 사항

- 보고서 제출 마감은 제출일 자정까지 (1시간 지연 허용)
- 지연 제출 시 감점 : 1일 지연 시 마다 30% 감점, 3일 지연 후부터는 미제출 처리
- 압축 오류, 파일 누락 관련 감점 syllabus 참고

○ 구현 점수

- 가. 헤더 파일 작성 6
- 나. q1.java를 c 프로그램으로 변환 10
- 다. q2.java를 c 프로그램으로 변환 20
- 라. q3.java를 c 프로그램으로 변환 20
- 마. -j 옵션 구현 5
- 바. -c 옵션 구현 5
- 사. -p 옵션 구현 8
- 아. -f 옵션 구현 5
- 자. -l 옵션 구현 5
- 차. -r 옵션 구현 8
- 카. c 파일에 각 라인에 맞게 탭 구성 8

○ 필수 구현 사항 : 가, 나, 다, 라, 마, 바, 아, 자