

<Function Interposition>

주어진 main.o funcs.o를 컴파일 하여 수행 하면 funcReplaced() 와 endFunc()이 호출된 숫자를 출력한다.(main.c 는 참조만 한다) 수행 파일의 인자로 funcReplaced 의 호출 횟수가 주어진다.

```
> ./a.out 3
```

```
In funcReplaced
```

```
In funcReplaced
```

```
In funcReplaced
```

```
In endFunc
```

```
funcReplaced is called 3 times
```

```
endFunc is called 1 times
```

1) Link-time interposition

funcReplaced()와 endFunc()을 link time에 다른 함수로 바꾸어 funcReplaced 몇 번 호출 되는지 파악하는데 사용되는 함수를 myfunc.c에 작성하라. 단 이때 원래 수행하던 프로그램이 제대로 수행되어야 한다.

다음 명령어를 사용하여 수행 파일 intl을 만들어 수행하면 다음과 같이 출력된다. (제공 파일 : main.o funcs.o)

```
> gcc -c myfunc.c; gcc -Wall -Wl,--wrap,funcReplaced -Wl,--wrap,endFunc -o intl main.o funcs.o myfunc.o
```

```
> ./intl 3
```

```
In funcReplaced
```

```
In wrap_funcReplaced
```

```
In funcReplaced
```

```
In wrap_funcReplaced
```

```
In funcReplaced
```

```
In wrap_funcReplaced
```

```
In endFunc
```

```
In wrap_endFunc
```

```
funcReplaced is called 3 times
```

```
endFunc is called 1 times
```

참고로 .o 파일의 symbol은 다음과 같은 명령어로 확인 할 수 있다.

```
> nm -g 파일이름.o
```

또는

```
> readelf -s 파일이름.o
```

2) Run-time interposition

funcReplaced()와 endFunc()을 run time에 다른 함수로 바꾸어 funcReplaced 몇 번 호출 되는지 파악하는데 사용되는 dynamic library를 만들 수 있는 **mydynamicfunc.c**에 작성한다. 단 이때 원래 수행하던 프로그램이 제대로 수행되어야 한다.

다음 명령어를 사용하여 수행 파일 intr을 만들어 수행하면 다음과 같이 출력된다. (제공 파일 : main.o libfuncs.so)

```
> gcc -Wall -shared -fpic -o mydynamicfunc.so mydynamicfunc.c -ldl
> gcc -L. -o intr main.o -lfuncs -Wl,-rpath, "."
> LD_PRELOAD="./mydynamicfunc.so" ./intr 3
In funcReplaced
In wrap_funcReplaced
In funcReplaced
In wrap_funcReplaced
In funcReplaced
In wrap_funcReplaced
In endFunc
In wrap_endFunc
funcReplaced is called 3 times
endFunc is called 1 times
```

또는

```
>gcc -Wall -shared -fpic -o mydynamicfunc.so mydynamicfunc.c -ldl
> gcc -L. -o intr main.o -lfuncs
> LD_LIBRARY_PATH="." LD_PRELOAD="./mydynamicfunc.so" ./intr 3
```

참고: ldd intr 또는 ldd 파일이름.so (dynamic library) 수행하면 해당 수행파일이나 dynamic library가 사용하는 library가 리스트 된다.

참고 : 필요하면 #define _GNU_SOURCE 대신 #define __USE_GNU을 사용하세요