



Korea

검색

| & | & | 가

Linux

DLL



plugin

[Allen Wilson \(wilsona@us.ibm.com\)](#)

e-business architect (IBM)

2001 10

Plugin DLL

business

. Linux

plugin DLL

Allen Wilson

가

. e-

[Linux](#)

[Linux dll](#)

[dll](#) : C [dITest](#)

[dITest](#)

[Listings \(dll\)](#)

가

dW :

[\(object](#)

[disoriented\)](#)

[US](#)

Also in the Linux zone:

[&](#)

[&](#)

plugin

. plugin

plugin

DLL

dll

backplan

dll
(packaging)

Linux

libc

. libc

가

. Linux

가

Linux

, plugin dll

Linux

Linux

가

:

(lib*.a)

(lib*.so)

가

Linux

:

```

main()
{
    printf("Hello world
");
}

```

```
gcc hello.c , a.out . Linux
ldd a.out , :
```

```
libc.so.6 => /lib/libc.so.6 (0x4001d000)
/lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
```

가 가 dll 가
Linux

Linux dll

Linux 가 (dlopen, dlerror, dlsym, dlclose),
include file (dlfcn.h), (library libdl.a, library libdl.so)

- dlopen
- dlsym
- dlerror NULL an ASCII
- dlclose

dlopen
가 :

- dlopen call
- LD_LIBRARY_PATH
- /etc/ld.so.cache
- /usr/lib /lib

dll : C dlTest C C
"Hello World" "HeLIO WoRID"

1. dll dlfcn.h 가 :
 -
 -
 -
2. "HeLIO WoRID"가
3. UPPERCASE dll dlopen absolute path
"/home/dlTest/UPPERCASE.so" RTLD_LAZY
 - RTLD_LAZY dll dll
 - Option RTLD_NOW dlopen dll
4. printUPPERCASE dlsym
5. printUPPERCASE "HELLO WORLD"가
6. UPPERCASE.so dlclose dll
7. lowercase dll lowercase.so dlopen
LD_LIBRARY_PATH

8. printLowercase dlsym .
9. printLowercase가 "hello world"가 .
10. lowercase.so dlclose dll .

dlopen, dlsym dlclose dlerror가 .
dlTest :

```

dlTest 2-Original message
HeLlO WoRlD
dlTest 3-Open Library with absolute path return-(null)-
dlTest 4-Find symbol printUPPERCASE return-(null)-
HELLO WORLD
dlTest 5-printUPPERCASE return-(null)-
dlTest 6-Close handle return-(null)-
dlTest 7-Open Library with relative path return-(null)-
dlTest 8-Find symbol printLowercase return-(null)-
hello world
dlTest 9-printLowercase return-(null)-
dlTest 10-Close handle return-(null)-

```

dlTest.c, UPPERCASE.c, lowercase.c

[Listings](#)

dlTest

1. dll
2. dll
3. dl

```

UPPERCASE.c lowercase.c gcc -fpic -fPIC
shared .UPPERCASE.o lowercase.o gcc
*.so
dltest ksh

```

```

#!/bin/ksh
# Build shared library
#
#set -x
clear

#
# Shared library for dlopen absolute path test
#
if [ -f UPPERCASE.o ]; then rm UPPERCASE.o
fi
gcc -c -fpic UPPERCASE.c
if [ -f UPPERCASE.so ]; then rm UPPERCASE.so
fi

```

```

gcc -shared -lc -o UPPERCASE.so UPPERCASE.o

#
# Shared library for dlopen relative path test
#
export LD_LIBRARY_PATH=`pwd`
if [ -f lowercase.o ]; then rm lowercase.o
fi
gcc -c -fpic lowercase.c
if [ -f lowercase.so ]; then rm lowercase.so
fi
gcc -shared -lc -o lowercase.so lowercase.o

#
# Rebuild test program
#
if [ -f dlTest ]; then rm dlTest
fi
gcc -o dlTest dlTest.c -ldl
echo Current LD_LIBRARY_PATH=$LD_LIBRARY_PATH
dlTest

```

Linux

dlopen, dlsym, dlclose

dlerror

relative

dll

dll

absolute path

LD_LIBRARY_PATH

path

가

-fpic

-fPIC

-shared

Linux

가

third-party plugin

Listings (

dll)

dlTest.c:

```

/*****
/*      Test Linux Dynamic Function Loading      */
/*      */
/*      void      *dlopen(const char *filename, int flag)      */
/*      Opens dynamic library and return handle      */
/*      */
/*      const char *dlerror(void)      */
/*      Returns string describing the last error.      */
/*      */
/*      void      *dlsym(void *handle, char *symbol)      */
/*      Return pointer to symbol's load point.      */
/*      If symbol is undefined, NULL is returned.      */
/*      */
/*      int      dlclose (void *handle)      */

```

```

/*          Close the dynamic library handle.          */
/*          */
/*          */
/*          */
/*****/
#include<stdio.h>
#include <stdlib.h>

/*          */
/* 1-dll include file and variables */
/*          */
#include <dlfcn.h>
void *FunctionLib;          /* Handle to shared lib file */
int (*Function)();          /* Pointer to loaded routine */
const char *dlError;          /* Pointer to error string */

main( argc, argv )
{
    int rc;          /* return codes */
    char HelloMessage[] = "HeLlO WoRlD\n";

/*          */
/* 2-print the original message */
/*          */
    printf("          dlTest 2-Original message \n");
    printf("%s", HelloMessage);

/*          */
/* 3-Open Dynamic Loadable Library with absolute path */
/*          */
    FunctionLib = dlopen("/home/dlTest/UPPERCASE.so",RTLD_LAZY);
    dlError = dlerror();
    printf("          dlTest 3-Open Library with absolute path return-%s- \n", dlError);
    if( dlError ) exit(1);

/*          */
/* 4-Find the first loaded function */
/*          */
    Function = dlsym( FunctionLib, "printUPPERCASE");
    dlError = dlerror();
    printf("          dlTest 4-Find symbol printUPPERCASE return-%s- \n", dlError);
    if( dlError ) exit(1);

/*          */
/* 5-Execute the first loaded function */
/*          */
    rc = (*Function)( HelloMessage );
    printf("          dlTest 5-printUPPERCASE return-%s- \n", dlError);

/*          */
/* 6-Close the shared library handle */
/* Note: after the dlclose, "printUPPERCASE" is not loaded */
/*          */
    rc = dlclose(FunctionLib);
    dlError = dlerror();
    printf("          dlTest 6-Close handle return-%s-\n",dlError);

```

```

    if( rc ) exit(1);

/*
/* 7-Open Dynamic Loadable Library using LD_LIBRARY path
/*
FunctionLib = dlopen("lowercase.so",RTLD_LAZY);
dlError = dlerror();
printf("      dlTest 7-Open Library with relative path return-%s- \n", dlError);
if( dlError ) exit(1);

/*
/* 8-Find the second loaded function
/*
Function = dlsym( FunctionLib, "printLowercase");
dlError = dlerror();
printf("      dlTest 8-Find symbol printLowercase return-%s- \n", dlError);
if( dlError ) exit(1);

/*
/* 8-execute the second loaded function
/*
rc = (*Function)( HelloMessage );
printf("      dlTest 9-printLowercase return-%s- \n", dlError);

/*
/* 10-Close the shared library handle
/*
rc = dlclose(FunctionLib);
dlError = dlerror();
printf("      dlTest 10-Close handle return-%s-\n",dlError);
if( rc ) exit(1);

return(0);
}

```

UPPERCASE.c:

```

/*****
/*      Function to print input string as UPPER case.
/*      Returns 1.
/*****

int printUPPERCASE ( inLine )
char inLine[];
{
    char UPstring[256];
    char *inptr, *outptr;

    inptr = inLine;
    outptr = UPstring;
    while ( *inptr != '\0' )
        *outptr++ = toupper(*inptr++);
    *outptr++ = '\0';
}

```

```
printf(UPstring);
return(1);
}
```

lowercase.c

```

/*****
/*      Function to print input string as lower case.      */
/*      Returns 2.                                          */
/***** */
int printLowercase( inLine )
char inLine[];
{
    char lowstring[256];
    char *inptr, *outptr;
    inptr = inLine;
    outptr = lowstring;
    while ( *inptr != ' ' )
        *outptr++ = tolower(*inptr++);
    *outptr++ = ' ';
    printf(lowstring);
    return(2);
}

```

- [Linux application tools](#) (zSeries & S/390):
- [IBM developer kit for Linux](#) :
- *developerWorks* : "[Shared objects for the object disoriented!](#)".
- [Linux resources](#) : *developerWorks*.
- [Open source resources](#) : *developerWorks*.

Allen Wilson (wilsona@us.ibm.com) IBM e-business , Linux, AIX,
 WebSphere Application Server, 가 .

?

- (5) (4) (3) (2) (1)



| | |