Sourcecode: Example3.c

COLLABORATORS			
	TITLE : Sourcecode: Example3.c		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		February 12, 2023	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

Sourcecode: Example3.c

1.1 Example3.c

```
/*********************************************************/
/*                                                       */
/* Amiga C Encyclopedia (ACE)           Amiga C Club (ACC) */
/* --------------------------           ------------------ */
/*                                                       */
/* Manual:  AmigaDOS                    Amiga C Club       */
/* Chapter: Introduction                Tulevagen 22       */
/* File:    Example3.c                  181 41  LIDINGO    */
/* Author:  Anders Bjerin               SWEDEN             */
/* Date:    93-09-24                                     */
/* Version: 1.1                                          */
/*                                                       */
/*   Copyright 1993, Anders Bjerin - Amiga C Club (ACC)  */
/*                                                       */
/* Registered members may use this program freely in their */
/*     own commercial/noncommercial programs/articles.   */
/*                                                       */
/*********************************************************/

/* For experienced users only! This example demonstrates how */
/* to create a long word aligned BPTR. The actual BPTR (the  */
/* memory used to store the BPTR address in) is long word    */
/* aligned. This is rarely needed since you normally only    */
/* work with addresses to data blocks which have to be long  */
/* word aligned. However, if you ever have to give AmigaDOS  */
/* an actual BPTR (not a BPTR address, but the BPTR itself)  */
/* you need to allocate it as described in this example.     */



/* Some advanced (and probably confusing) information about BCPL    */
/* pointers:                                                        */
/*                                                                  */
/* When you are working with AmigaDOS you will often use variables  */
/* which have been declared as BPTRs (BPTR and BSTR are defined in  */
/* header file "dos/dos.h"). What you have to remember is that      */
/* these pointers must point to long word aligned data and must be  */
/* in BPCL form (four times smaller than normal C pointers).        */
```

```
/*                                                                  */
/* To decalre a BPTR simply write:                                  */
/*                                                                  */
/*   BPTR my_bcpl_pointer;                                          */
/*                                                                  */
/* When you work with AmigaDOS you will often call functions, for   */
/* example Open(), which will return a BPTR (a BPCL address). The   */
/* memory which this returned BPCL pointer points to will have      */
/* been allocated by the function itsef and will therefore be long  */
/* word aligned. The address (value) in the BPTR will therefore     */
/* point to long word aligned memory, and can be used with          */
/* functions which requires BCPL pointers.                          */
/*                                                                  */
/* Now comes the tricky part! The pointer itself (the memory used   */
/* to store the address in) is NOT long word aligned when you       */
/* declare it as described above. Normally this is not a problem    */
/* since you usually only work with the address stored in the BPTR. */
/* However, if you ever would have to give AmigaDOS a BPCL pointer  */
/* (not the address in the pointer, but the pointer itself) you     */
/* must make sure that the actual pointer (the memory used to store */
/* the addresses in) is also long word aligned.                     */



/* Include the normal dos header file: */
#include <libraries/dos.h>

/* Include memory definitions: (MEMF_ANY...) */
#include <exec/memory.h>

/* Now we include the necessary function prototype files:         */
#include <clib/dos_protos.h>       /* General dos functions...    */
#include <clib/exec_protos.h>      /* System functions...         */
#include <stdio.h>                 /* Std functions [printf()...] */
#include <stdlib.h>                /* Std functions [exit()...]   */



/* Set name and version number: */
UBYTE *version = "$VER: AmigaDOS/AmigaDOS/Example3 1.0";



/* Declared our own function(s): */
int main( int argc, char *argv[] );



/* The main function: */

int main( int argc, char *argv[] )
{
  /* Declare a normal C pointer to the */
  /* BPTR pointer we will allocate:    */
  LONG *my_aligned_bptr;
```

```
  /* Allocate some memory for the aligned BPTR pointer: */
  /* (The memory we allocate will be long word aligned. */
  /* We allocate 4 bytes = 1 long.)                     */
  my_aligned_bptr = AllocMem( sizeof( BPTR ),
    MEMF_ANY | MEMF_CLEAR );


  /* Have we successfully allocated the memory? */
  if( !my_aligned_bptr )
  {
    /* Not enough memory! Inform the user and quit: */
    printf( "Could not allocate enough memory!\n" );

    /* Exit with an error code: */
    exit( 20 );
  }


  /* We have now allocated a long word aligned BPTR! */
  /* Note that the "my_aligned_bptr" contains the C  */
  /* address of the aligned BPTR pointer!            */

  /* You can now use the long word aligned BPTR...   */
  printf( "We can now use the long word aligned BPTR!\n" );



  /* Deallocate the long word aligned BPTR when you */
  /* do not need it any more:                       */
  FreeMem( my_aligned_bptr, sizeof( BPTR ) );

  /* Remember that you may not use the memory any */
  /* more after you have deallocated it!          */
  printf( "The long word aligned BPTR has been deallocated!\n" );

  /* The End! */
  exit( 0 );
}
```