```
/*********************************************************************
Copyright 2010-2017 K.C. Wang, <kwang@eecs.wsu.edu>
This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program.  If not, see <http://www.gnu.org/licenses/>.
*********************************************************************/

/*********************************************************************
                       io.c file of MTX
*********************************************************************/
char space = ' ';
char *ctable = "0123456789ABCDEF";
char cr = '\r';

int puts(const char *s){ }

#define printk printf

int printf(char *fmt,...);

typedef struct ext2_dir_entry_2 {
        u32     inode;                  /* Inode number */
        u16     rec_len;                /* Directory entry length */
        u8      name_len;               /* Name length */
        u8      file_type;
        char    name[255];              /* File name */
} DIR;

typedef struct stat {
  u16    st_dev;               /* major/minor device number */
  u16    st_ino;               /* i-node number */
  u16    st_mode;              /* file mode, protection bits, etc. */
  u16    st_nlink;             /* # links; TEMPORARY HACK: should be nlink_t*/
  u16    st_uid;                       /* uid of the file's owner */
  u16    st_gid;               /* gid; TEMPORARY HACK: should be gid_t */
  u16    st_rdev;
  long   st_size;              /* file size */
  long   st_atime;             /* time of last access */
  long   st_mtime;             // time of last modification
  long   st_ctime;             // time of creation
  long   st_dtime;
  long   st_date;
  long   st_time;
} STAT;


// UNIX <fcntl.h> constants: <asm/fcntl.h> in Linux
#define O_RDONLY             00
#define O_WRONLY             01
#define O_RDWR               02
#define O_CREAT            0100 /* not fcntl */
#define O_TRUNC            01000 /* not fcntl */
#define O_APPEND           02000
```

```c
#define EOF  -1

#define exit mexit
/*
#define O_RDONLY 0
#define O_WRONLY 1
#define O_RDWR   2
*/
int mputc(char c)
{
   write(1, &c, 1);
   if (c=='\n')
     write(1,&cr,1);
   return 0;
}


void prints(char *s)
{
   while (*s){
      mputc(*s);
      s++;
   }
}

void mputs(char *s)
{
  prints(s);
}



//void align(), printi(), prints();
/****************
int strcmp(char *s1, char *s2)
{
  while(*s1 && *s2){
    if (*s1 != *s2)
      return *s1 - *s2;
    s1++; s2++;
  }
  return 0;
}
***************/

/********
char getc()
{
  return syscall(11,0,0,0);
}

void gets(char *s)
{   char c; int len=0;

    while ( (c=getc()) != '\r' && len < 64){
           *s++ = c; len++;
           mputc(c);
    }
    prints("\n\r");
    *s = 0;
}
```

```
********/
extern int strlen(const char *);
void print2f(char *s)
{
  write(2, s, (int)strlen(s));
}
/*
void mptchar(char c)
{
  mptc(c);
  if (c=='\r')
    mptc('\n');
}
*/
/***************
void align(u32 x)
{
  int count;
  count = 6;
  if (x==0)
     count = 5;
  while (x){
    count--;
    x = (u32)(x/10);
  }

  while(count){
    mputc(space);
    count--;
  }
}
***************/


void rpi(int x)
{
   char c;
   if (x==0) return;
   c = ctable[x%10];
   rpi((int)x/10);
   mputc(c);
}

void printi(int x)
{
    if (x==0){
       prints("0 ");
       return;
    }
    if (x < 0){
       mputc('-');
       x = -x;
    }
    rpi((int)x);
    mputc(space);
}

void rpu(u32 x)
{
   char c;
   if (x==0) return;
   c = ctable[x%10];
```

```
   rpi((u32)x/10);
   mputc(c);
}

void printu(u32 x)
{
    if (x==0){
       prints("0 ");
       return;
    }
    rpu((u32)x);
    mputc(space);
}

void rpx(u32 x)
{
   char c;
   if (x==0) return;
   c = ctable[x%16];
   rpx((u32)x/16);
   mputc(c);
}

void printx(u32 x)
{
  prints("0x");
   if (x==0){
      prints("0 ");
      return;
   }
   rpx((u32)x);
  mputc(space);
}


void printc(char c)
{
  mputc(c);
  c = c&0x7F;
  if (c=='\n')
    mputc(cr);
}

int printk(char *fmt,...)
{
  char *cp, *cq;
  int  *ip;

  cq = cp = (char *)fmt;
  ip = (int *)&fmt + 1;

  while (*cp){
    if (*cp != '%'){
       printc(*cp);
       cp++;
       continue;
    }
    cp++;
    switch(*cp){
      case 'd' : printi(*ip); break;
      case 'u' : printu(*ip); break;
      case 'x' : printx(*ip); break;
```

```
      case 's' : prints((char *)*ip); break;
      case 'c' : printc((char)*ip);   break;
    }
    cp++; ip++;
  }
}
```