Initiation à la programmation en C

Correction du TP n°6

Antoine Miné

22 mars 2007

Site du cours: `http://www.di.ens.fr/~mine/enseignement/prog2006/`

Exercice 1. Commande `wc`.

version à un argument

```
#include <stdio.h>
#include <string.h>

/* Note: la dernière ligne d'un fichier n'est pas comptée si elle ne se
   termine pas par un retour à la ligne
*/
void compte_lignes(const char* nom_fichier)
{
  FILE* f = fopen( nom_fichier, "r" ); /* flux ouvert en lecture */
  char ligne[1024];                    /* pour stocker la ligne à lire */
  int l = 0;                           /* nombre de lignes lues */

  /* échec de l'ouverture */
  if ( !f ) { printf( "impossible d'ouvrir %s\n", nom_fichier ); return; }

  while (1) {
    /* lit une ligne */
    if (fgets( ligne, sizeof(ligne)-1, f ) == NULL)
      break; /* fin de fichier */

    /* incrémente le compteur seulement si la ligne lue se termine bien par
       un retour à la ligne
    */
    if (ligne[strlen(ligne)-1] == '\n') l++;
  }

  printf( "%s: %i lignes\n", nom_fichier, l );

  fclose(f);
}


int main(int argc, char* argv[])
{
  if (argc != 2) fprintf( stderr, "Donnez-moi un nom de fichier!\n" );
  else compte_lignes( argv[1] );
  return 0;
}
```

version à plusieurs arguments

```c
#include <stdio.h>
#include <string.h>

int total = 0; /* nombre total de lignes lues */

void compte_lignes(const char* nom_fichier)
{
  FILE* f = fopen( nom_fichier, "r" );
  char ligne[1024];
  int l = 0;
  if ( !f ) { printf( "impossible d'ouvrir %s\n", nom_fichier ); return; }
  while (1) {
    if (fgets( ligne, sizeof(ligne)-1, f ) == NULL) break;
    if (ligne[strlen(ligne)-1] == '\n') { l++; total++; }
  }
  printf( "%s: %i lignes\n", nom_fichier, l );
  fclose(f);
}


int main(int argc, char* argv[])
{
  int i;

  /* pour chaque argument en ligne de commande */
  for ( i=1; i<argc; i++ ) compte_lignes( argv[i] );

  printf( "\ntotal: %i lignes\n", total );
  return 0;
}
```

Exercice 2. Lecture d'un entier.

```c
#include <stdio.h>

int fgetint(FILE* stream)
{
  int acc = 0; /* accumulateur */

  while (1) {

    /* lit un caractère */
    int c = getc(stream);

    /* vérifie que c'est un chiffre */
    if ( c < '0' || c > '9' ) break;

    /* ajoute le chiffre à droite de l'accumulateur */
    acc = acc * 10 + (c - '0');
  }

  return acc;
}

/* programme de test */
int main()
{
  int x = fgetint( stdin );
  printf( "Vous avez tapé %i\n", x );
```

```
  return 0;
}
```

version acceptant les négatifs

```
int fgetint(FILE* stream)
{
  int signe = 1;         /* signe: +1 ou -1 */
  int acc = 0;           /* accumulateur de la valeur absolue */
  int c = getc(stream);  /* mange le premier caractère */

  /* cas où le premier caractère est '+' ou '-'  */
  if ( c == '-' )      { signe = -1; c = getc(stream); }
  else if ( c == '+' ) { c = getc(stream); }

  while ( c>='0' && c<='9' ) {
    acc = acc * 10 + (c - '0'); /* ajoute un chiffre */
    c = getc(stream);           /* mange le prochain caractère */
  }

  return acc * signe;
}
```

Exercice 3. Manipulation d'images au format `.ppm` binaire.

```
#include <stdlib.h>
#include <stdio.h>
#include <errno.h>
#include <string.h>

#define MAX 1024 /* taille maximale de l'image */

unsigned char img[MAX*MAX*3]; /* tableau global contenant l'image */

int l,h;    /* taille de l'image */

void charge_image(const char* fichier, int* largeur, int* hauteur,
                  unsigned char* image)
{
  FILE* f = fopen(fichier, "r");
  char buf[256];
  int intensite, l, c;
  if (!f) { perror("charge_image"); exit(1); }
  fgets(buf, 255, f);
  if (strcmp(buf,"P6\n")) { printf("mauvais format"); exit(1); }
  fgets(buf, 255, f);
  fscanf(f, "%i %i %i", largeur, hauteur, &intensite);
  if (*largeur > MAX || *hauteur > MAX) { printf("image trop grande"); exit(1); }
  for ( l=0; l<*hauteur; l++ )
    for ( c=0; c<*largeur; c++ ) {
      image[ ((*largeur*l)+c)*3    ] = fgetc(f);
      image[ ((*largeur*l)+c)*3 +1 ] = fgetc(f);
      image[ ((*largeur*l)+c)*3 +2 ] = fgetc(f);
    }
  fclose(f);
}

void sauve_image(const char* fichier, int largeur, int hauteur,
                 unsigned char* image)
{
```

```
  FILE* f = fopen(fichier, "w");
  int l, c;
  if (!f) { perror("sauve_image"); exit(1); }
  fprintf(f, "P6\n# hello\n%i %i\n%i\n", largeur, hauteur, 255);
  for ( l=0; l<hauteur; l++ )
    for ( c=0; c<largeur; c++ ) {
      fputc( image[ ((largeur*l)+c)*3    ], f);
      fputc( image[ ((largeur*l)+c)*3 +1 ], f);
      fputc( image[ ((largeur*l)+c)*3 +2 ], f);
    }
  fclose(f);
}

void reduit_image(int largeur, int hauteur, unsigned char* image)
{
  int x,y,i;
  for ( y=0; y<hauteur; y++ )
    for ( x=0; x<largeur; x++ )
      for ( i=0; i<3; i++ )
        image[ ((largeur/2*y)+x)*3 + i ] =
          ( image[ ((largeur*(2*y  ))+(2*x  ))*3 + i ] +
            image[ ((largeur*(2*y+1))+(2*x  ))*3 + i ] +
            image[ ((largeur*(2*y  ))+(2*x+1))*3 + i ] +
            image[ ((largeur*(2*y+1))+(2*x+1))*3 + i ] ) / 4;
}


int main(int argc, char*argv[])
{
  if (argc!=3) printf("donnez-moi des arguments!\n");
  else {
    charge_image(argv[1], &l, &h, img);
    reduit_image(l, h, img);
    sauve_image(argv[2], l/2, h/2, img);
  }
  return 0;
}
```