mini\_ls\_couleursFIchiers.c

#include <stdio.h>  
#include <sys/stat.h>  
#include <pwd.h>  
#include <grp.h>  
#include <string.h>  
#include <sys/types.h>  
#include <time.h>  
#include <unistd.h>  
#include <sys/sysmacros.h>  
#include <limits.h>  
  
int main(int argc, char \* argv[]) {  
 struct stat statbuf, sb;  
 int retour\_stat, i = 0, j = 0;  
 char fichier\_source[PATH\_MAX \* 2 +5] = ""; *// 2 fois la longueur maximale + de l'espace pour la flèche*  
char type[100];  
  
 if (argc == 1) {  
 printf("Usage: %s fichier1 [fichier2 ...]\n", argv[0]);  
 return 1;  
 }  
 for (i = 1; i < argc; i++) {  
 struct passwd \* pw = NULL;  
 struct group \* gr = NULL;  
 char permission[4] = "";  
 char droits[20] = "";  
 char Date\_modification[30] = "";  
  
 retour\_stat = lstat(argv[i], & statbuf);  
 if (retour\_stat == -1) {  
 perror("Erreur de la récupération des informations");  
 return 1;  
 }  
 switch (statbuf.st\_mode & S\_IFMT) {  
 case S\_IFDIR:  
 strcpy(type, "répertoire");  
 break;  
 case S\_IFLNK:  
 strcpy(type, "lien symbolique");  
 break;  
 case S\_IFREG:  
 strcpy(type, "fichier standard");  
 break;  
 default:  
 strcpy(type, "Autre");  
 break;  
 }  
 sprintf(permission, "%o", statbuf.st\_mode & 0777);  
 for (j = 0; j < 4; j++) {  
 switch (permission[j]) {  
 case '0':  
 strcat(droits, "---");  
 break;  
 case '1':  
 strcat(droits, "--x");  
 break;  
 case '2':  
 strcat(droits, "-w-");  
 break;  
 case '3':  
 strcat(droits, "-wx");  
 break;  
 case '4':  
 strcat(droits, "r--");  
 break;  
 case '5':  
 strcat(droits, "r-x");  
 break;  
 case '6':  
 strcat(droits, "rw-");  
 break;  
 case '7':  
 strcat(droits, "rwx");  
 break;  
  
 }  
 }  
 pw = getpwuid(statbuf.st\_uid);  
 gr = getgrgid(statbuf.st\_gid);  
 strftime(Date\_modification, 30, "%b %e %H:%M", localtime( & statbuf.st\_mtime));  
  
 if (i == 1) {  
 printf("| Numéro inode | Droits | Type | Nombre de lien durs | Utilisateur | Groupe | Taille | Dernière modification | fichier \n");  
 }  
 printf("| %-12ld | %-11s | %-16s | %-19ld | %-11s | %-6s | %-11ld | %-21s |",  
 statbuf.st\_ino, droits, type, statbuf.st\_nlink, pw -> pw\_name, gr->gr\_name, statbuf.st\_size, Date\_modification);  
 if (strcmp(type, "répertoire") == 0) {  
 printf("\033[1;34m");  
 printf(" %s \n", argv[i]);  
 printf("\033[0m");  
 } else if (strcmp(type, "fichier standard") == 0 && (statbuf.st\_mode & 0111)) {  
 printf("\033[1;32m");  
 printf(" %s \n", argv[i]);  
 printf("\033[0m");  
 } else if (strcmp(type, "lien symbolique") == 0) {  
 int rl = readlink(argv[i], fichier\_source, statbuf.st\_size + 1);  
 if (rl == -1) {  
 perror("Erreur en obtenant les infos sur le lien symbolique");  
 return 1;  
 }  
 if (stat(argv[i], & sb) == -1) {  
 printf("\033[1;31m");  
 printf(" %s ", argv[i]);  
 printf("\033[0m");  
 printf("->");  
 printf("\033[1;31m");  
 printf(" %s\n", fichier\_source);  
 printf("\033[0m");  
 } else {  
 printf("\033[1;36m");  
 printf(" %s ", argv[i]);  
 printf("\033[0m");  
 printf("-> %s\n", fichier\_source);  
 }  
 } else {  
 printf(" %s \n", argv[i]);  
 }  
 }  
 return 0;  
}

mini\_ls\_fichierspointes.c

#include <stdio.h>  
#include <sys/stat.h>  
#include <stdlib.h>  
#include <pwd.h>  
#include <grp.h>  
#include <string.h>  
#include <sys/types.h>  
#include <time.h>  
#include <unistd.h>  
#include <sys/sysmacros.h>  
  
int main(int argc, char\* argv[]){  
 struct stat statbuf;  
 int retour\_stat, i=0, j=0;  
 char lien\_sym[30]="";  
 if(argc == 1){  
 printf("Usage: %s fichier1 [fichier2 ...]\n", argv[0]);  
 return 1;  
 }  
 for (i=1; i<argc; i++){  
 struct passwd \*pw = NULL;  
 struct group \*gr = NULL;  
 char permission[4]="";  
 char droits[20]="";  
 char Date\_modification[30]="";  
 char nom\_fichier[100];  
 retour\_stat = lstat(argv[i], &statbuf);  
 if(retour\_stat == -1){  
 perror("Erreur de la récupération des informations");  
 return 1;  
 }  
 strcpy(nom\_fichier, argv[i]) ;  
  
 if(readlink(argv[i], lien\_sym, statbuf.st\_size+1) != -1){  
 strcat(nom\_fichier," -> ");  
 strcat(nom\_fichier, lien\_sym);  
 }  
 sprintf(permission, "%o", statbuf.st\_mode & 0777);  
 for(j=0; j<4; j++){  
 switch (permission[j]){  
 case '0':  
 strcat(droits,"---");  
 break;  
 case '1':  
 strcat(droits,"--x");  
 break;  
 case '2':  
 strcat(droits,"-w-");  
 break;  
 case '3':  
 strcat(droits,"-wx");  
 break;  
 case '4':  
 strcat(droits,"r--");  
 break;  
 case '5':  
 strcat(droits,"r-x");  
 break;  
 case '6':  
 strcat(droits,"rw-");  
 break;  
 case '7':  
 strcat(droits,"rwx");  
 break;  
 }  
 }  
 pw = getpwuid(statbuf.st\_uid);  
 gr = getgrgid(statbuf.st\_gid);  
 strftime(Date\_modification, 30, "%b %e %H:%M", localtime(&statbuf.st\_mtime));  
 if(i == 1){  
 printf("| Numéro inode | Droits | Nombre de lien durs | Utilisateur | Groupe | Taille | Dernière modification | fichier \n");  
 }  
 printf("| %-12ld | %-11s | %-19ld | %-11s | %-6s | %-11ld | %-21s | %s \n", statbuf.st\_ino, droits, statbuf.st\_nlink, pw->pw\_name, gr->gr\_name, statbuf.st\_size, Date\_modification, nom\_fichier);  
 }  
 return 0;

}

mini\_ls\_plusieurFichiers.c

#include <stdio.h>  
#include <sys/stat.h>  
#include <fcntl.h>  
#include <pwd.h>  
#include <grp.h>  
#include <string.h>  
#include <sys/types.h>  
#include <time.h>  
#include <sys/sysmacros.h>  
  
int main(int argc, char\* argv[]){  
 struct stat statbuf;  
 int retour\_stat, i, j;  
 if(argc == 1){  
 printf("Usage: %s fichier1 [fichier2 ...]\n", argv[0]);  
 return 1;  
 }  
 for (i=1; i<argc; i++){  
 struct passwd \*pw = NULL;  
 struct group \*gr = NULL;  
 char type[30];  
 char permission[4]="";  
 char droits[20]="";  
 char Date\_modification[30]="";  
 retour\_stat = lstat(argv[i], &statbuf);  
 if(retour\_stat == -1){  
 perror("Erreur de la récupération des informations");  
 return 1;  
 }  
 switch (statbuf.st\_mode & S\_IFMT) {  
 case S\_IFDIR: strcpy(type, "répertoire"); break;  
 case S\_IFLNK: strcpy(type, "lien symbolique"); break;  
 case S\_IFREG: strcpy(type, "fichier standard"); break;  
 default: strcpy(type, "Autre"); break;  
 }  
 if(i == 1){  
 printf("| Numéro inode | Type | Droits | Nombre de lien durs | Utilisateur | Groupe | Taille | Dernière modification \n");  
 }  
 sprintf(permission, "%o", statbuf.st\_mode & 0777);  
 for(j=0; j<4; j++){  
 switch (permission[j]){  
 case '0':  
 strcat(droits,"---");  
 break;  
 case '1':  
 strcat(droits,"--x");  
 break;  
 case '2':  
 strcat(droits,"-w-");  
 break;  
 case '3':  
 strcat(droits,"-wx");  
 break;  
 case '4':  
 strcat(droits,"r--");  
 break;  
 case '5':  
 strcat(droits,"r-x");  
 break;  
 case '6':  
 strcat(droits,"rw-");  
 break;  
 case '7':  
 strcat(droits,"rwx");  
 break;  
 }  
 }  
  
 pw = getpwuid(statbuf.st\_uid);  
 gr = getgrgid(statbuf.st\_gid);  
 strftime(Date\_modification, 30, "%b %e %H:%M", localtime(&statbuf.st\_mtime));  
 printf("| %-12ld | %-20s | %-11s | %-19ld | %-11s | %-6s | %-11ld | %-29s \n", statbuf.st\_ino, droits, type, statbuf.st\_nlink, pw->pw\_name, gr->gr\_name, statbuf.st\_size, Date\_modification);  
 }  
 return 0;  
}

mini\_ls\_uid\_gid.c

#include <stdio.h>  
#include <sys/stat.h>  
#include <unistd.h>  
#include <sys/types.h>  
#include <string.h>  
#include <time.h>  
#include <sys/sysmacros.h>  
int main(int argc, char \* argv[]) {  
 struct stat statbuf;  
 int retour\_stat, i;  
 char type[30];  
 char permission[4];  
 char droits[20] = "";  
 char Date\_modification[30];  
 if (argc != 2) {  
 printf("Usage: %s nom\_du\_fichier\n", argv[0]);  
 return 1;  
 }  
 retour\_stat = lstat(argv[1], & statbuf);  
 if (retour\_stat == -1) {  
 perror("Erreur de la récupération des informations");  
 return 1;  
 }  
 switch (statbuf.st\_mode & S\_IFMT) {  
 case S\_IFDIR:  
 strcpy(type, "répertoire");  
 break;  
 case S\_IFLNK:  
 strcpy(type, "lien symbolique");  
 break;  
 case S\_IFREG:  
 strcpy(type, "fichier standard");  
 break;  
 default:  
 strcpy(type, "Autre");  
 break;  
 }  
 sprintf(permission, "%o", statbuf.st\_mode & 0777);  
 for (i = 0; i < 4; i++) {  
 switch (permission[i]) {  
 case '0':  
 strcat(droits, "---");  
 break;  
 case '1':  
 strcat(droits, "--x");  
 break;  
 case '2':  
 strcat(droits, "-w-");  
 break;  
 case '3':  
 strcat(droits, "-wx");  
 break;  
 case '4':  
 strcat(droits, "r--");  
 break;  
 case '5':  
 strcat(droits, "r-x");  
 break;  
 case '6':  
 strcat(droits, "rw-");  
 break;  
 case '7':  
 strcat(droits, "rwx");  
 break;  
 }  
 }  
 strftime(Date\_modification, 30, "%b %e %H:%M", localtime( & statbuf.st\_mtime));  
 printf("| Numéro inode | Type | Droits | Nombre de lien durs | uid | gid | Taille | Dernière modification \n");  
 printf("| %-12ld | %-20s | %-11s | %-19ld | %-8d | %-8d | %-11ld | %-29s \n", statbuf.st\_ino, type, droits, statbuf.st\_nlink, statbuf.st\_uid, statbuf.st\_gid, statbuf.st\_size, Date\_modification);  
 return 0;  
}