

Linux Modification

**Jeremy Goldberg, Andrey Varakin,
Miliano Mikol**

Motivation

```
Type | User | Group | World | Link Count | Owner | Associated Group | Size(byte  
s) | Last Modified | Name  
- rw- r-- 1 whynot1597 whynot1597 92K Apr 20 19:50 ABOUT-NLS
```

- Already quite familiar with “ls” when we started the project
- Whole point of “ls-lh” is to be clear and readable, however it was still difficult to tell what some of the information being displayed was
- Decided to add our own flag to the module to add headers for each type of data displayed, as well as extra spacing between the permission flags



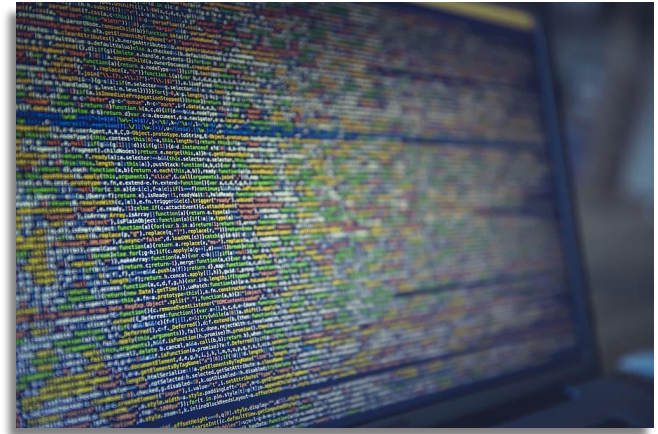
Beginning Steps and Plan

- Figuring out where `ls.c` was located
- Wanted to maintain functionality of the existing program
- Scoured the 5600+ line file to understand the module
- Edit necessary files (`ls.c`, `filemode.c`, header files)



Problems Encountered During Coding

1. Coreutils files can be long!!
2. Building strings in C is not as easy as other languages
3. In modifying a large product, there are a lot of dependencies



Additions

```
432
433 // ADDED
434 static bool extra_readable;
435
```

```
838 /* For long options that have no equivalent short option, use a
839 non-character as a pseudo short option, starting with CHAR_MAX + 1. */
840 enum
841 {
842     AUTHOR_OPTION = CHAR_MAX + 1,
843     BLOCK_SIZE_OPTION,
844     COLOR_OPTION,
845     DEREFERENCE_COMMAND_LINE_SYMLINK_TO_DIR_OPTION,
846     FILE_TYPE_INDICATOR_OPTION,
847     FORMAT_OPTION,
848     FULL_TIME_OPTION,
849     GROUP_DIRECTORIES_FIRST_OPTION,
850     HIDE_OPTION,
851     HYPERLINK_OPTION,
852     INDICATOR_STYLE_OPTION,
853     QUOTING_STYLE_OPTION,
854     // ADDED
855     READABLE_OPTION,
856     SHOW_CONTROL_CHARS_OPTION,
857     SI_OPTION,
858     SORT_OPTION,
859     TIME_OPTION,
860     TIME_STYLE_OPTION
861 };
862
```

```
874 {"kibibytes", no_argument, NULL, 'k'},
875 {"numeric-uid-gid", no_argument, NULL, 'n'},
876 {"no-group", no_argument, NULL, 'G'},
877 {"hide-control-chars", no_argument, NULL, 'q'},
878 {"reverse", no_argument, NULL, 'r'},
879 {"size", no_argument, NULL, 's'},
880 {"width", required_argument, NULL, 'w'},
881 // ADDED
882 {"display", no_argument, NULL, READABLE_OPTION},
883 {"almost-all", no_argument, NULL, 'A'},
884 {"ignore-backups", no_argument, NULL, 'B'},
885 {"classify", no_argument, NULL, 'F'},
886 {"file-type", no_argument, NULL, FILE_TYPE_INDICATOR_OPTION},
887 {"si", no_argument, NULL, SI_OPTION},
888 {"dereference-command-line", no_argument, NULL, 'H'},
```

```
2182
2183 // ADDED
2184 case READABLE_OPTION:
2185     format = long_format;
2186     file_human_output_opts = human_output_opts =
2187         human_autoscale | human_SI | human_base_1024;
2188     file_output_block_size = output_block_size = 1;
2189     extra_readable = true;
2190     break;
```

```
4089 case long_format:
4090     // ADDED
4091     if (extra_readable)
4092     {
4093         printf("Type | User | Group | World | Link Count | Owner | Associated\n");
4094     }
4095
4096     for (i = 0; i < cwd_n_used; i++)
4097     {
4098         set_normal_color();
4099         print_long_format(sorted_file[i]);
4100         DIRED_PUTCHAR('\n');
4101     }
4102     break;
```

Additions (cont.)

```
4247 /* Print information about F in long format. */
4248 static void
4249 print_long_format(const struct fileinfo *f)
4250 {
4251     // ADDED
4252     // more accurately - change size of modebuf from 12 to 15
4253     char modebuf[15];
4254     char buf
4255         [LONGEST_HUMAN_READABLE + 1 /* inode */
4256          + LONGEST_HUMAN_READABLE + 1 /* size in blocks */
4257          + sizeof(modebuf) - 1 + 1 /* mode string */
4258          + INT_BUFSIZE_BOUND(uintmax_t) /* st_nlink */
4259          + LONGEST_HUMAN_READABLE + 2 /* major device number */
4260          + LONGEST_HUMAN_READABLE + 1 /* minor device number */
4261          + TIME_STAMP_LEN_MAXIMUM + 1 /* max length of time/date */
4262     ];
4263     size_t s;
4264     char *p;
4265     struct timespec when_timespec;
4266     struct tm when_local;
4267     bool btime_ok = true;
4268
4269     /* Compute the mode string, except remove the trailing space if no
4270      file in this directory has an ACL or security context. */
4271     if (f->stat_ok)
4272         // ADDED extra_readable flag
4273         // Check filemode.c for more additions
4274         filemodestring(&f->stat, modebuf, extra_readable);
4275 }
```

```
166 void filemodestring(struct stat const *statp, char *str, bool extra_readable)
167 {
168     strmode(statp->st_mode, str);
169     if (extra_readable)
170     {
171         char col_str[15];
172         for (int i = 0, j = 0; j < 10; i++, j++)
173         {
174             col_str[i] = str[j];
175             if (j % 3 == 0)
176             {
177                 i++;
178                 col_str[i] = ' ';
179             }
180         }
181         col_str[14] = '\0';
182         memcpy(str, col_str, sizeof(char) * 15);
183     }
```