

```
1: // $Id: binarydump.c,v 1.1 2016-10-13 12:52:01-07 - - $
2:
3: //
4: // Dump out files in binary.
5: //
6:
7: #include <ctype.h>
8: #include <errno.h>
9: #include <fcntl.h>
10: #include <libgen.h>
11: #include <stdio.h>
12: #include <stdlib.h>
13: #include <string.h>
14: #include <unistd.h>
15:
16: char *execname = NULL;
17: int exit_status = EXIT_SUCCESS;
18:
19: void syserror (char *filename) {
20:     exit_status = EXIT_FAILURE;
21:     fflush (NULL);
22:     fprintf (stderr, "%s: %s: %s\n",
23:             execname, filename, strerror (errno));
24:     fflush (NULL);
25: }
26:
27: void binary_dump (char *filename, int in_fdes) {
28:     printf ("%s:\n", filename);
29:     char buffer[8];
30:     ssize_t offset = 0;
31:     for (;;) {
32:         ssize_t nbytes = read (in_fdes, buffer, sizeof buffer);
33:         if (nbytes < 0) syserror (filename);
34:         if (nbytes <= 0) break;
35:         printf ("%5zd", offset);
36:         offset += nbytes;
37:         for (ssize_t ichar = 0; ichar < nbytes; ++ichar) {
38:             printf (" ");
39:             for (int bit = 0x80; bit != 0; bit >>= 1) {
40:                 printf ("%d", (buffer[ichar] & bit) != 0);
41:             }
42:         }
43:         printf ("\n");
44:         printf ("%5s", "");
45:         for (ssize_t ichar = 0; ichar < nbytes; ++ichar) {
46:             char byte = buffer[ichar];
47:             printf (" %03o %02X %c",
48:                 byte, byte, isgraph (byte) ? byte : ' ');
49:         }
50:         printf ("\n");
51:     }
52:     printf ("%5zd\n", offset);
53: }
54:
```

```
55:
56: int main (int argc, char **argv) {
57:     execname = basename (argv[0]);
58:     if (argc == 1) {
59:         binary_dump ("-", STDIN_FILENO);
60:     }else {
61:         for (int argi = 1; argi < argc; ++argi) {
62:             char *filename = argv[argi];
63:             if (strcmp (filename, "-") == 0) {
64:                 binary_dump ("-", STDIN_FILENO);
65:             }else {
66:                 int in_fdes = open (filename, O_RDONLY);
67:                 if (in_fdes < 0) {
68:                     syserror (filename);
69:                 }else {
70:                     binary_dump (filename, in_fdes);
71:                     int rc = close (in_fdes);
72:                     if (rc < 0) syserror (filename);
73:                 }
74:             }
75:         }
76:     }
77:     return exit_status;
78: }
```

```
1: // $Id: hexadecimaldump.c,v 1.1 2016-10-13 12:52:01-07 - - $
2:
3: //
4: // Dump out files in binary.
5: //
6:
7: #include <ctype.h>
8: #include <errno.h>
9: #include <fcntl.h>
10: #include <libgen.h>
11: #include <stdio.h>
12: #include <stdlib.h>
13: #include <string.h>
14: #include <unistd.h>
15:
16: char *execname = NULL;
17: int exit_status = EXIT_SUCCESS;
18:
19: void syserror (char *filename) {
20:     exit_status = EXIT_FAILURE;
21:     fflush (NULL);
22:     fprintf (stderr, "%s: %s: %s\n",
23:             execname, filename, strerror (errno));
24:     fflush (NULL);
25: }
26:
27: void hex_dump (char *filename, int in_fdes) {
28:     printf ("%s:\n", filename);
29:     ssize_t offset = 0;
30:     for (;;) {
31:         char buffer[16];
32:         ssize_t nbytes = read (in_fdes, buffer, sizeof buffer);
33:         if (nbytes < 0) syserror (filename);
34:         if (nbytes == 0) break;
35:         printf ("%05zX", offset);
36:         offset += nbytes;
37:         for (ssize_t ichar = 0; ichar < 16; ++ichar) {
38:             if (ichar % 4 == 0) printf (" ");
39:             if (ichar < nbytes) {
40:                 printf ("%02X", (unsigned char) buffer[ichar]);
41:             } else {
42:                 printf (" ");
43:             }
44:         }
45:         printf (" |");
46:         for (ssize_t ichar = 0; ichar < 16; ++ichar) {
47:             if (ichar < nbytes) {
48:                 char byte = buffer[ichar];
49:                 printf ("%c", isprint (byte) ? byte : '.');
50:             } else {
51:                 printf (" ");
52:             }
53:         }
54:         printf ("|\n");
55:     }
56: }
```

```
57:
58:
59: int main (int argc, char **argv) {
60:     execname = basename (argv[0]);
61:     if (argc == 1) {
62:         hex_dump ("-", STDIN_FILENO);
63:     }else {
64:         for (int argi = 1; argi < argc; ++argi) {
65:             char *filename = argv[argi];
66:             if (strcmp (filename, "-") == 0) {
67:                 hex_dump ("-", STDIN_FILENO);
68:             }else {
69:                 int in_fdes = open (filename, O_RDONLY);
70:                 if (in_fdes < 0) {
71:                     syserror (filename);
72:                 }else {
73:                     hex_dump (filename, in_fdes);
74:                     int rc = close (in_fdes);
75:                     if (rc < 0) syserror (filename);
76:                 }
77:             }
78:         }
79:     }
80:     return exit_status;
81: }
82:
```