

```
1: // $Id: binarydump.c,v 1.3 2014-10-21 20:01:37-07 - - $
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 ("%7ld", 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 ("%7s", "");
45:         for (ssize_t ichar = 0; ichar < nbytes; ++ichar) {
46:             char byte = buffer[ichar];
47:             printf (" %03o %02X", byte, byte);
48:             if (isgraph (byte)) printf (" %c", byte);
49:             else printf (" ");
50:         }
51:         printf ("\n");
52:     }
53:     printf ("%7ld\n", offset);
54: }
55:
```

```
56:
57: int main (int argc, char **argv) {
58:     execname = basename (argv[0]);
59:     if (argc == 1) {
60:         binary_dump ("-", STDIN_FILENO);
61:     }else {
62:         for (int argi = 1; argi < argc; ++argi) {
63:             char *filename = argv[argi];
64:             if (strcmp (filename, "-") == 0) {
65:                 binary_dump ("-", STDIN_FILENO);
66:             }else {
67:                 int in_fdes = open (filename, O_RDONLY);
68:                 if (in_fdes < 0) {
69:                     syserror (filename);
70:                 }else {
71:                     binary_dump (filename, in_fdes);
72:                     int rc = close (in_fdes);
73:                     if (rc < 0) syserror (filename);
74:                 }
75:             }
76:         }
77:     }
78:     return exit_status;
79: }
```

```
1: // $Id: hexadecimaldump.c,v 1.1 2014-10-21 20:11:40-07 - - $
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 ("%07lX", 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:                 if (isprint (byte)) printf ("%c", byte);
50:                 else printf (".");
51:             } else {
52:                 printf (" ");
53:             }
54:         }
55:         printf ("|\n");
56:     }
57: }
```

```
58:
59:
60: int main (int argc, char **argv) {
61:     execname = basename (argv[0]);
62:     if (argc == 1) {
63:         hex_dump ("-", STDIN_FILENO);
64:     }else {
65:         for (int argi = 1; argi < argc; ++argi) {
66:             char *filename = argv[argi];
67:             if (strcmp (filename, "-") == 0) {
68:                 hex_dump ("-", STDIN_FILENO);
69:             }else {
70:                 int in_fdes = open (filename, O_RDONLY);
71:                 if (in_fdes < 0) {
72:                     syserror (filename);
73:                 }else {
74:                     hex_dump (filename, in_fdes);
75:                     int rc = close (in_fdes);
76:                     if (rc < 0) syserror (filename);
77:                 }
78:             }
79:         }
80:     }
81:     return exit_status;
82: }
83:
```