



*Full credit is given to the above companies including the OS that this PDF file was generated!*

## ***Red Hat Enterprise Linux Release 9.2 Manual Pages on 'a64l.3p' command***

***\$ man a64l.3p***

A64L(3P)                    POSIX Programmer's Manual                    A64L(3P)

### PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

### NAME

a64l, l64a ? convert between a 32-bit integer and a radix-64 ASCII string

### SYNOPSIS

```
#include <stdlib.h>

long a64l(const char *s);

char *l64a(long value);
```

### DESCRIPTION

These functions maintain numbers stored in radix-64 ASCII characters.

This is a notation by which 32-bit integers can be represented by up to six characters; each character represents a digit in radix-64 notation.

If the type long contains more than 32 bits, only the low-order 32 bits shall be used for these operations.

The characters used to represent digits are '.' (dot) for 0, '/' for 1, '0' through '9' for [2,11], 'A' through 'Z' for [12,37], and 'a' through 'z' for [38,63].

The a64l() function shall take a pointer to a radix-64 representation,

in which the first digit is the least significant, and return the corresponding long value. If the string pointed to by `s` contains more than six characters, `a64l()` shall use the first six. If the first six characters of the string contain a null terminator, `a64l()` shall use only characters preceding the null terminator. The `a64l()` function shall scan the character string from left to right with the least significant digit on the left, decoding each character as a 6-bit radix-64 number. If the type `long` contains more than 32 bits, the resulting value is sign-extended. The behavior of `a64l()` is unspecified if `s` is a null pointer or the string pointed to by `s` was not generated by a previous call to `l64a()`.

The `l64a()` function shall take a `long` argument and return a pointer to the corresponding radix-64 representation. The behavior of `l64a()` is unspecified if value is negative.

The value returned by `l64a()` may be a pointer into a static buffer.

Subsequent calls to `l64a()` may overwrite the buffer.

The `l64a()` function need not be thread-safe.

## RETURN VALUE

Upon successful completion, `a64l()` shall return the long value resulting from conversion of the input string. If a string pointed to by `s` is an empty string, `a64l()` shall return 0L.

The `l64a()` function shall return a pointer to the radix-64 representation. If value is 0L, `l64a()` shall return a pointer to an empty string.

## ERRORS

No errors are defined.

The following sections are informative.

## EXAMPLES

None.

## APPLICATION USAGE

If the type `long` contains more than 32 bits, the result of `a64l(l64a(x))` is `x` in the low-order 32 bits.

## RATIONALE

This is not the same encoding as used by either encoding variant of the

uuencode utility.

## FUTURE DIRECTIONS

None.

## SEE ALSO

strtoul()

The Base Definitions volume of POSIX.1?2017, <stdlib.h>

The Shell and Utilities volume of POSIX.1?2017, uuencode

## COPYRIGHT

Portions of this text are reprinted and reproduced in electronic form from IEEE Std 1003.1-2017, Standard for Information Technology -- Portable Operating System Interface (POSIX), The Open Group Base Specifications Issue 7, 2018 Edition, Copyright (C) 2018 by the Institute of Electrical and Electronics Engineers, Inc and The Open Group. In the event of any discrepancy between this version and the original IEEE and The Open Group Standard, the original IEEE and The Open Group Standard is the referee document. The original Standard can be obtained online at <http://www.opengroup.org/unix/online.html> .

Any typographical or formatting errors that appear in this page are most likely to have been introduced during the conversion of the source files to man page format. To report such errors, see [https://www.kernel.org/doc/man-pages/reporting\\_bugs.html](https://www.kernel.org/doc/man-pages/reporting_bugs.html) .

IEEE/The Open Group

2017

A64L(3P)