527326-13-1E AID: 258164| 27/06/2020

Directories are special in the sense that within a directory you can have references to several files and directories, so, if you remove the parent directory, all those files lose their reference point from where they can be accessed, the same with process. For such cases, rmdir() have different checks, that are different from unlink():

If thedirectory is not empty. If a directory is not empty it can't remove it until the contents are unlink'd/removed. If the directory is in use. If a process losses their current directory, it could lead to problems and undefined behaviors. Is better to prevent them.

In the case of unlink() these checks doesn't exist. In fact, you can delete the name of a file with unlink() and the process that is still using/making reference to it, can modify it without problems. The file exist until the file descriptor exist, just unaccessible to new process (unless you know where to search). This is part of the rainbow-colored-hands magic of the \*NIX file systems.

Now, there's the unlinkat() which behaves as both, unlink() or rmdir() depending the path which is what you expect.