Description of QNX-OS

Wikipedia:

QNX is a commercial Unix-like real-time operating system, aimed primarily at the embedded systems market. The product was originally developed in the early 1980s by Canadian company Quantum Software Systems, later renamed QNX Software Systems and ultimately acquired by BlackBerry in 2010.QNX was one of the first commercially successful microkernel operating systems[citation needed] and is used in a variety of devices including cars and mobile phones.

As a microkernel-based OS, QNX is based on the idea of running most of the operating system kernel in the form of a number of small tasks, known as servers. This differs from the more traditional monolithic kernel, in which the operating system kernel is a single very large program composed of a huge number of "parts" with special abilities. In the case of QNX, the use of a microkernel allows users (developers) to turn off any functionality they do not require without having to change the OS itself; instead, those servers will simply not run.

The system is quite small, with earlier versions fitting on a single floppy disk.

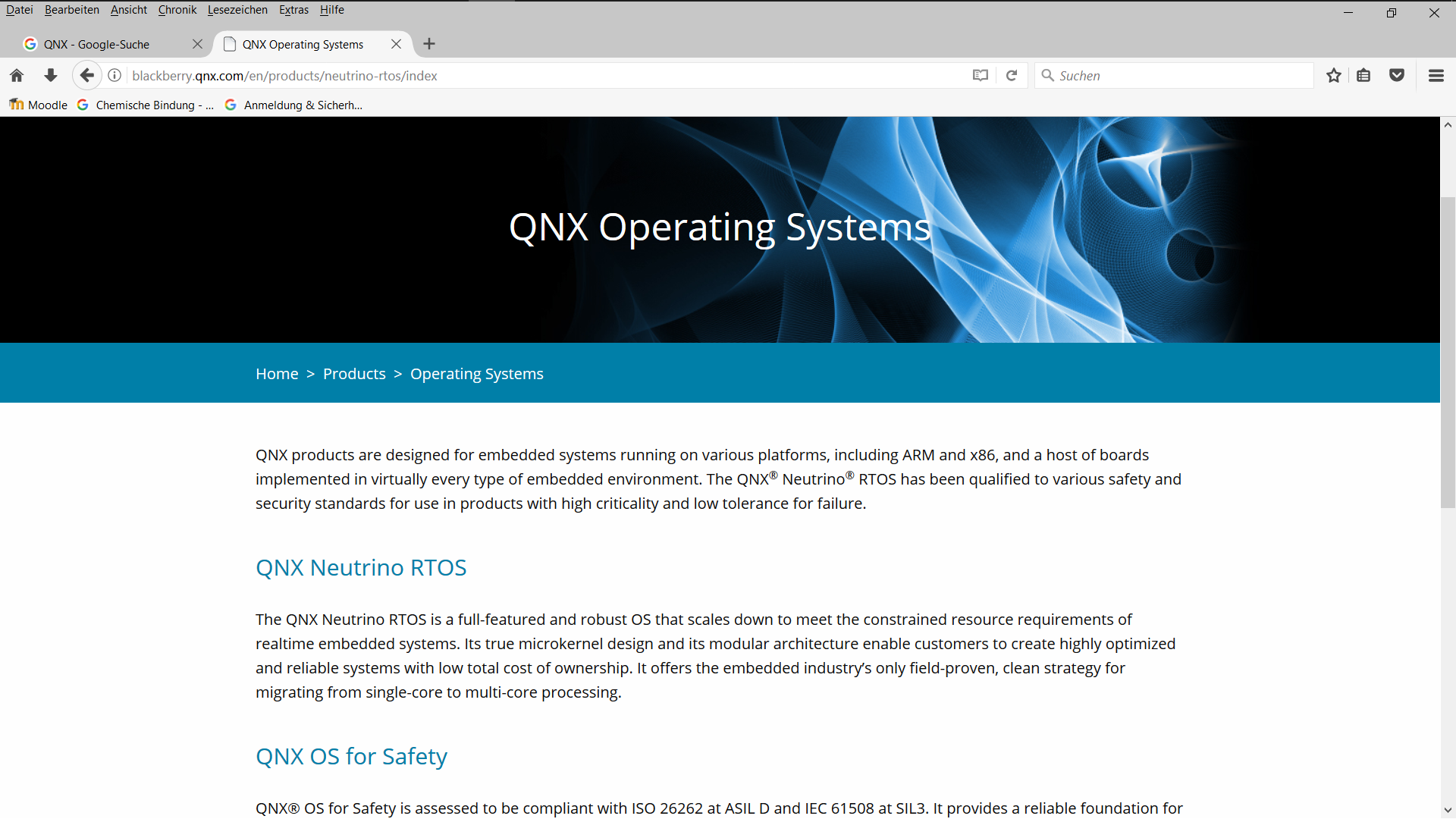
QNX Neutrino (2001) has been ported to a number of platforms and now runs on practically any modern CPU that is used in the embedded market. This includes the PowerPC, x86 family, MIPS, SH-4, and the closely inter-related family of ARM, StrongARM and XScale CPUs.

QNX offers a license for non-commercial and academic users.

The BlackBerry PlayBook tablet computer designed by BlackBerry uses a version of QNX as the primary operating system. Devices from BlackBerry running the BlackBerry 10 operating system are also based on QNX.

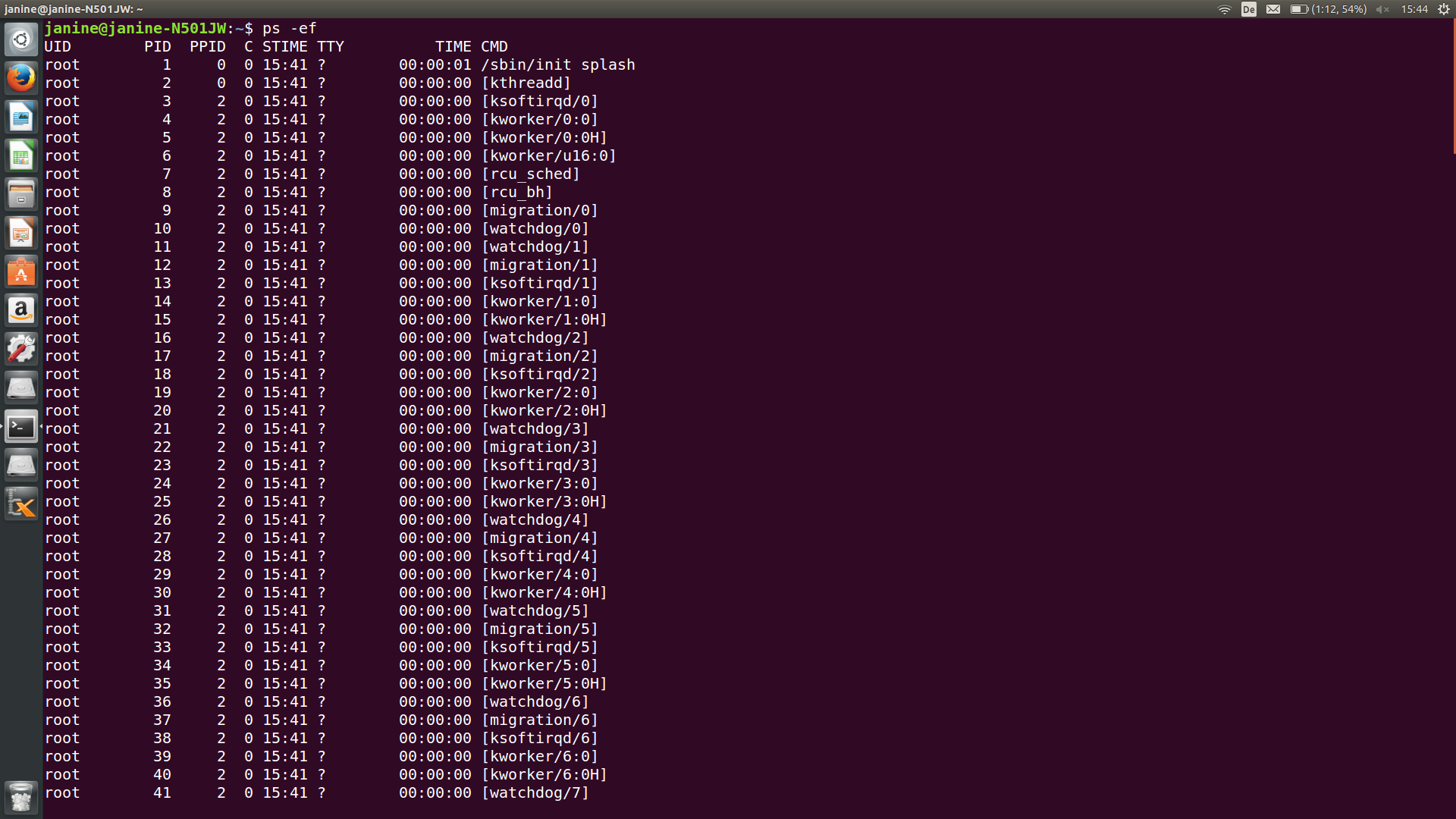
The QNX kernel contains only CPU scheduling, interprocess communication, interrupt redirection and timers. Everything else runs as a user process, including a special process known as proc which performs process creation and memory management by operating in conjunction with the microkernel. This is made possible by two key mechanisms — subroutine-call type interprocess communication, and a boot loader which can load an image containing not only the kernel but any desired collection of user programs and shared libraries. There are no device drivers in the kernel. The network stack is based on NetBSD code.[20] Along with its support for its own, native, device drivers, QNX supports its legacy, io-net manager server, and the network drivers ported from NetBSD.

Homepage of QNX:

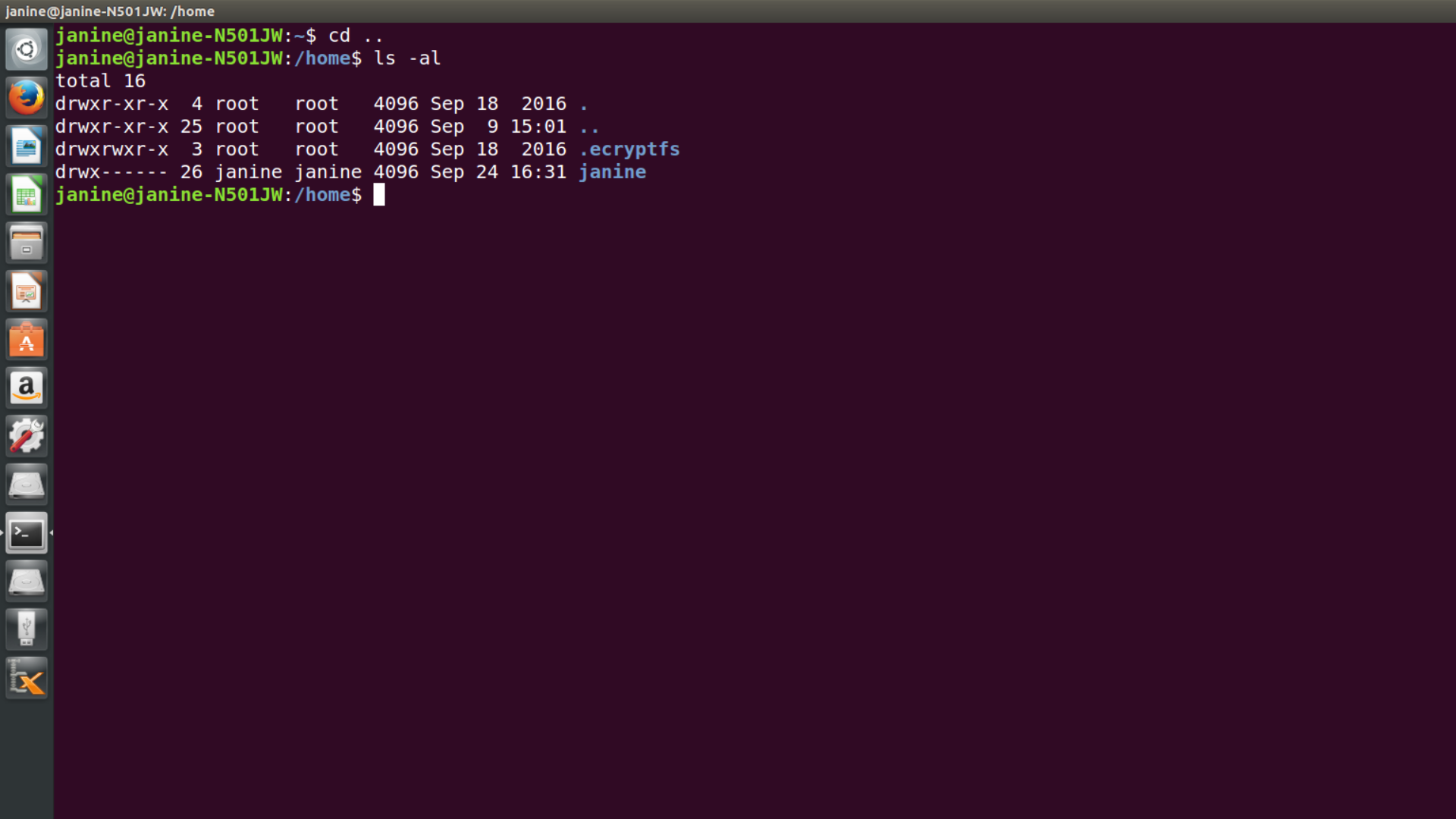


Required Tasks (Screenshots)

1. All processes “ps -ef”



1. Root directory: “ls -al”



1. Result “pwd” after issuing the command “cd (for example: Templates)

