The Operating System

"QNX Neutrino is a microkernel operating system that makes fault-resilient, field-upgradable systems much easier to design and implement. QNX Neutrino also offers unprecedented scalability: Developers can build their applications once, then distribute them on a single processor, across a massive cluster of processors, or on an SMP system - all without special coding. QNX Neutrino represents the latest generation of QNX OS technology, and is the product of more than 20 years' experience in the embedded market."

A conventional executive

A monolithic OS

A microkernel


Making an Image

A minimum buildfile.

```
[virtual=x86,bios +compress] .bootstrap = {
  startup-bios -s 64k
  PATH=/proc/boot procnto
}
[+script] .script = {
  display_msg "Hello"
}
```

Simple case in which one or more programs need to be used.

```
[virtual=x86,bios +compress] .bootstrap = {
  startup-bios -s 64k
  PATH=/proc/boot procnto
}
[+script] .script = {
  devc-tcon &
  reopen /dev/con1
  [+session] esh
}
[type=link] /usr/lib/ldqnx.so.2=/proc/boot/libc.so
libc.so
[data=copy]
devc-tcon
esh
ls
cat
sin
pidin
```
Instead of devc-tcon, devc-con can be used. This full console driver offers several consoles.

```forth
[virtual=x86,bios +compress] .bootstrap = {
    startup-bios -s 64k
    PATH=/proc/boot procnto
}
[+script] startup-script = {
    devc-con -n9 &
    reopen /dev/con1
    [+session] sh &
    reopen /dev/con2
    [+session] sh &
    reopen /dev/con3
    [+session] sh &
    reopen /dev/con4
    [+session] sh &
    reopen /dev/con5
    [+session] sh &
    reopen /dev/con6
    [+session] sh &
    reopen /dev/con7
    [+session] sh &
    reopen /dev/con8
    [+session] sh &
    reopen /dev/con9
    [+session] sh &
}
[type=link] /tmp=/dev/shmem
[type=link] /usr/lib/ldqnx.so.2=/proc/boot/libc.so
libc.so
[data=c]
devc-con
sh
ls
cat
pidin
shutdown
```
The next step can be introducing the network

```plaintext
[virtual=x86,bios +compress] .bootstrap = {
    startup-bios -s 64k
    PATH=/proc/boot procnto
}
[+script] startup-script = {
    devc-tcon &
    seedres
    pci-bios &
    waitfor /dev/pci
    io-net -d el900 -p ttcpip if=en0:150.0.1.44:255.255.255.0
    default=150.0.1.3
    reopen /dev/con1
    [+session] sh &
}
[type=link] /usr/lib/ldqnx.so.2=/proc/boot/libc.so
devn-el900.so
libc.so
libsocket.so
npm-ttcpip.so
[data=copy]
devc-con
io-net
pci-bios
seedres
sh
cat
ls
ftp
ping
telnet
```
A buildfile for accessing the floppy disk

```plaintext
[virtual=x86,bios +compress] .bootstrap = {
    startup-bios -s 64k
    PATH=/proc/boot procnto
}
[+script] startup-script = {
    seedres
    pci-bios &
    waitfor /dev/pci
    devb-fdc cam quite disk name=fd blk
        automount=fd0:/qnx4,cache=64k
    devc-tcon &
    reopen /dev/con1
[+session] sh
}
[type=link] /usr/lib/ldqnx.so.2=/proc/boot/libc.so
    cam-disk.so
    fs-qnx4.so
    io-blk.so
    libc.so
    libcam.so
[+data=copy]
    devb-fdc
    devc-tcon
    pci-bios
    seedres
    sh
    ...
```
A buildfile for accessing the hard disk

```plaintext
[virtual=x86,bios +compress] .bootstrap = {
    startup-bios -s 64k
    PATH=/proc/boot procnto
}
[+script] startup-script = {
    seedres
    pci-bios &
    waitfor /dev/pci
    devb-eide cam quite blk automount=hd0t77:/qnx4
    devc-tcon &
    reopen /dev/con1
    [+session] sh
}
[type=link] /usr/lib/ldqnx.so.2=/proc/boot/libc.so
    cam-disk.so
    fs-qnx4.so
    io-blk.so
    libc.so
    libcam.so
    [data=copy]
    devb-eide
    devc-tcon
    pci-bios
    seedres
    sh
    ...
```
Using Forth as a working environment

[virtual=x86,bios +compress] .bootstrap = {
    startup-bios -s 64k
    PATH=/proc/boot procnto
}
[+script] .script = {
    devc-tcon &
    waitfor /dev/con1
    forth
}
[type=link] /usr/lib/ldqnx.so.2=/proc/boot/libc.so
libc.so
devc-tcon
forth

<table>
<thead>
<tr>
<th>Offset</th>
<th>Size</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>400</td>
<td>*.boot</td>
</tr>
<tr>
<td>400</td>
<td>100</td>
<td>Startup-header flags1=0xd flags2=0 paddr_bias=0</td>
</tr>
<tr>
<td>500</td>
<td>a008</td>
<td>startup.*</td>
</tr>
<tr>
<td>a508</td>
<td>5c</td>
<td>Image-header mountpoint=/</td>
</tr>
<tr>
<td>a564</td>
<td>198</td>
<td>Image-directory</td>
</tr>
<tr>
<td>----</td>
<td>----</td>
<td>Root-dirent</td>
</tr>
<tr>
<td>----</td>
<td>12</td>
<td>usr/lib/ldqnx.so.2 -&gt; /proc/boot/libc.so</td>
</tr>
<tr>
<td>a6fc</td>
<td>44</td>
<td>proc/boot/.script</td>
</tr>
<tr>
<td>b000</td>
<td>36368</td>
<td>proc/boot/procnto</td>
</tr>
<tr>
<td>42000</td>
<td>4d000</td>
<td>proc/boot/libc.so.2</td>
</tr>
<tr>
<td>----</td>
<td>9</td>
<td>proc/boot/libc.so -&gt; libc.so.2</td>
</tr>
<tr>
<td>8f000</td>
<td>9000</td>
<td>proc/boot/devc-tcon</td>
</tr>
<tr>
<td>98000</td>
<td>6000</td>
<td>proc/boot/forth</td>
</tr>
</tbody>
</table>

Checksums: image=0x2ed0254f startup=0xa104f52d