What: /sys/firmware/memmap/
Date: June 2008
Contact: Bernhard Walle <bernhard.walle@gmx.de>
Description:

On all platforms, the firmware provides a memory map which the kernel reads. The resources from that memory map are registered in the kernel resource tree and exposed to userspace via /proc/iomem (together with other resources).

However, on most architectures that firmware-provided memory map is modified afterwards by the kernel itself, either because the kernel merges that memory map with other information or just because the user overwrites that memory map via command line.

kexec needs the raw firmware-provided memory map to setup the parameter segment of the kernel that should be booted with kexec. Also, the raw memory map is useful for debugging. For that reason, /sys/firmware/memmap is an interface that provides the raw memory map to userspace.

The structure is as follows: Under /sys/firmware/memmap there are subdirectories with the number of the entry as their name:

```
/sys/firmware/memmap/0
/sys/firmware/memmap/1
/sys/firmware/memmap/2
/sys/firmware/memmap/3
...
```

The maximum depends on the number of memory map entries provided by the firmware. The order is just the order that the firmware provides.

Each directory contains three files:

start : The start address (as hexadecimal number with the '0x' prefix).
end : The end address, inclusive (regardless whether the firmware provides inclusive or exclusive ranges).
type : Type of the entry as string. See below for a list of valid types.

So, for example:

```
/sys/firmware/memmap/0/start
/sys/firmware/memmap/0/end
/sys/firmware/memmap/0/type
/sys/firmware/memmap/1/start
...
```

Currently following types exist:

- System RAM
- ACPI Tables
- ACPI Non-volatile Storage

- reserved

Following shell snippet can be used to display that memory
map in a human-readable format:

```
-------------------- 8< ----------------------------------------
  #!/bin/bash
  cd /sys/firmware/memmap
  for dir in * ; do
      start=$(cat $dir/start)
      end=$(cat $dir/end)
      type=$(cat $dir/type)
      printf "%016x-%016x (%s)\n" $start $[ $end +1] "$type"
  done
-------------------- >8 ----------------------------------------
```