uvesafb - A Generic Driver for VBE2+ compliant video cards

1. Requirements

uvesafb should work with any video card that has a Video BIOS compliant with the VBE 2.0 standard.

Unlike other drivers, uvesafb makes use of a userspace helper called v86d. v86d is used to run the x86 Video BIOS code in a simulated and controlled environment. This allows uvesafb to function on arches other than x86. Check the v86d documentation for a list of currently supported arches.

v86d source code can be downloaded from the following website:

https://github.com/mjanusz/v86d

Please refer to the v86d documentation for detailed configuration and installation instructions.

Note that the v86d userspace helper has to be available at all times in order for uvesafb to work properly. If you want to use uvesafb during early boot, you will have to include v86d into an initramfs image, and either compile it into the kernel or use it as an initrd.

2. Caveats and limitations

uvesafb is a _generic_ driver which supports a wide variety of video cards, but which is ultimately limited by the Video BIOS interface. The most important limitations are:

- Lack of any type of acceleration.
- A strict and limited set of supported video modes. Often the native or most optimal resolution/refresh rate for your setup will not work with uvesafb, simply because the Video BIOS doesn't support the video mode you want to use. This can be especially painful with widescreen panels, where native video modes don't have the 4:3 aspect ratio, which is what most BIOS-es are limited to.
- Adjusting the refresh rate is only possible with a VBE 3.0 compliant Video BIOS. Note that many nVidia Video BIOS-es claim to be VBE 3.0 compliant, while they simply ignore any refresh rate settings.

3. Configuration

uvesafb can be compiled either as a module, or directly into the kernel. In both cases it supports the same set of configuration options, which are either given on the kernel command line or as module parameters, e.g.:

```
video=uvesafb:1024x768-32,mtrr:3,ywrap (compiled into the kernel)

# modprobe uvesafb mode_option=1024x768-32 mtrr=3 scroll=ywrap  (module)
```

Accepted options:

ypan	Enable display panning using the VESA protected mode interface. The visible screen is just a window of the video memory, console scrolling is done by changing the start of the window. This option is available on x86 only and is the default option on that architecture.
ywrap	Same as ypan, but assumes your gfx board can wrap-around the video memory (i.e. starts reading from top if it reaches the end of video memory). Faster than ypan. Available on x86 only.
redraw	Scroll by redrawing the affected part of the screen, this is the default on non-x86.

(If you're using uvesafb as a module, the above three options are used a parameter of the scroll option, e.g. scroll=ypan.)

vgapal	Use the standard VGA registers for palette changes.
pmipal	Use the protected mode interface for palette changes. This is the default if the protected mode interface is available. Available on x86 only.
mtrr:n	Setup memory type range registers for the framebuffer where n: • 0 - disabled (equivalent to nomtrr) • 3 - write-combining (default) Values other than 0 and 3 will result in a warning and will be treated just like 3.
nomtrr	Do not use memory type range registers.
vremap:n	Remap 'n' MiB of video RAM. If 0 or not specified, remap memory according to video mode.
vtotal:n	If the video BIOS of your card incorrectly determines the total amount of video RAM, use this option to override the BIOS (in MiB).

<mode>	The mode you want to set, in the standard modedb format. Refer to modedb.txt for a detailed description. When uvesafb is compiled as a module, the mode string should be provided as a value of the 'mode_option' option.
vbemode:x	Force the use of VBE mode x. The mode will only be set if it's found in the VBE-provided list of supported modes. NOTE: The mode number 'x' should be specified in VESA mode number notation, not the Linux kernel one (eg. 257 instead of 769). HINT: If you use this option because normal <mode> parameter does not work for you and you use a X server, you'll probably want to set the 'nocrtc' option to ensure that the video mode is properly restored after console <-> X switches.
nocrtc	Do not use CRTC timings while setting the video mode. This option has any effect only if the Video BIOS is VBE 3.0 compliant. Use it if you have problems with modes set the standard way. Note that using this option implies that any refresh rate adjustments will be ignored and the refresh rate will stay at your BIOS default (60 Hz).
noedid	Do not try to fetch and use EDID-provided modes.
noblank	Disable hardware blanking.
v86d:path	Set path to the v86d executable. This option is only available as a module parameter, and not as a part of the video= string. If you need to use it and have uvesafb built into the kernel, use uvesafb.v86d="path".

Additionally, the following parameters may be provided. They all override the EDID-provided values and BIOS defaults. Refer to your monitor's specs to get the correct values for maxhf, maxvf and maxclk for your hardware.

maxhf:n	Maximum horizontal frequency (in kHz).
maxvf:n	Maximum vertical frequency (in Hz).
maxclk:n	Maximum pixel clock (in MHz).

4. The sysfs interface

uvesafb provides several sysfs nodes for configurable parameters and additional information.

Driver attributes:

/sys/bus/platform/drivers/uvesafb

v86d

(default: /sbin/v86d)

Path to the v86d executable. v86d is started by uvesafb if an instance of the daemon isn't already running.

Device attributes:

/sys/bus/platform/drivers/uvesafb/uvesafb.0

nocrtc
: Use the default refresh rate (60 Hz) if set to 1.

oem_product_name, oem_product_rev, oem_string, oem_vendor
: Information about the card and its maker.

vbe_modes
: A list of video modes supported by the Video BIOS along with their VBE mode numbers in hex.

vbe_version
: A BCD value indicating the implemented VBE standard.

5. Miscellaneous

Uvesafb will set a video mode with the default refresh rate and timings from the Video BIOS if you set pixclock to 0 in fb_var_screeninfo.

Michal Januszewski <spock@gentoo.org>

Last updated: 2017-10-10

Documentation of the uvesafb options is loosely based on vesafb.txt.