

# Re: What File System supports Application XIP

**Source:** <http://linux.derkeiler.com/Mailing-Lists/Kernel/2004-09/3053.html>

---

**From:** Tim Bird ([tim.bird\\_at\\_am.sony.com](mailto:tim.bird_at_am.sony.com))

**Date:** 09/09/04

Date: Thu, 09 Sep 2004 09:35:38 -0700

To: colin <[colin@realtek.com.tw](mailto:colin@realtek.com.tw)>

David Woodhouse wrote:

> *On Thu, 2004-09-09 at 10:19 +0100, Paulo Marques wrote:*  
>> *colin wrote:*  
>>>  
>>> *Hi there,*  
>>> *We are developing embedded Linux system. Performance is our consideration.*  
>>> *We hope some applications can run as fast as possible,*  
>>> *and are think if they can be put in a filesystem image, which resides in*  
>>> *RAM, and run in XIP (eXecute In Place) manners.*  
>>> *I know that Cramfs has supported Application XIP. Is there any other FS that*  
>>> *also supports it? Ramdisk? Ramfs? Romfs?*  
>>>  
>>> *Obviously cramfs can not support XIP, because the "in-place" image*  
>>> *is compressed (unless you have a processor that can execute compressed*  
>>> *code :)*  
>>>  
>>> *Actually there are hacks floating around which do let you use XIP with*  
>>> *cramfs -- obviously you have to dispense with compression for the files*  
>>> *you want to access that way.*

The patches I've seen require setting the CRAMFS\_LINEAR option, to turn on linear addressing for cramfs, and CRAMFS\_LINEAR\_XIP. The result of these is to dispense with compression.

>  
> *You won't gain at runtime by using XIP though. Your code and data will*  
> *end up in RAM \_whatever\_ file system you use, and you'll be running from*  
> *page cache.*

Really? I thought that when you XIPed from flash, the page was mapped directly back to the flash memory address. There shouldn't be a RAM copy of the page at all.

Greg Ungerer (uClinux maintainer) once wrote:

> *[Application XIP provides a] win of keeping the application code in*  
> *flash even when that is shared. Can make a difference on small memory*  
> *systems.*

## Linux–Kernel: Re: What File System supports Application XIP

- >
- > *It also helps alleviate the contiguous memory problem (or memory fragmentation if you prefer) when you don't have an MMU. We need to*
- > *be able to allocate a big enough contiguous memory region to load*
- > *the text into. Can be a problem on systems that have been running*
- > *for a while and free memory is fragmented. If the application can*
- > *be run XIP from flash then you at least don't need to worry about*
- > *that. (This is a very real problem on small RAM systems).*

In the CE Linux Forum, we've been investigating Application XIP for two purposes: reduction in RAM footprint, and faster startup time. However, those both come at the sacrifice of runtime performance (as David says below). Running XIP from a RAM–based filesystem, as originally proposed, might solve the performance problem, but it wouldn't improve space utilization, and since you have to populate the RAM from somewhere on startup anyway, I don't think it will improve your bootup time. It *might* improve per–application startup time, once the system is running, but I think a regular RAM disk suffices for that.

- > *You may get a \_slightly\_ faster startup time after reboot if you use XIP*
- > *from flash, because it doesn't have to be loaded into RAM first. But*
- > *that comes at the cost of making it all a low slower during normal*
- > *operation -- it's a lot slower to fetch icache lines from flash than it*
- > *is from RAM.*

FYI – Here are some rough numbers:

Time to run shell script which starts TinyX X server and "xsetroot –solid red", then shuts down:

First invocation: Non–XIP 3.195 seconds, XIP 2.035 seconds

Second invocation: Non–XIP 1.744 seconds, XIP 1.765 seconds

I think this was on a 133 MHz PPC, but I'm not positive. In both cases the filesystem was in flash. Note that once the application pages are in RAM in the page cache, the Non–XIP case beats the XIP case (probably due to the penalty to access flash memory).

So the only performance win is on the first invocation of the application.

I'm just now starting to put together a wiki page of information about Application XIP, for those interested. (It has nothing now, so don't get excited.) But contributions are welcome at:

<http://tree.celinuxforum.org/pubwiki/moin.cgi/ApplicationXIP>

(Maybe when someone googles this message a year from now, there will be something interesting there... ;–)

=====

Tim Bird

Architecture Group Co–Chair, CE Linux Forum

Senior Staff Engineer, Sony Electronics

Re: What File System supports Application XIP

Linux-Kernel: Re: What File System supports Application XIP

E-mail: tim.bird@am.sony.com

=====

—

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@vger.kernel.org

More majordomo info at <http://vger.kernel.org/majordomo-info.html>

Please read the FAQ at <http://www.tux.org/lkml/>