

Linux



Complete Debug
from Kernel to Applications

Debug Embedded Linux by **PARTNER-Jet**

You can debug virtually everything, from the Linux kernel to applications, using a JTAG ICE, PARTNER-Jet and a debug software, the PARTNER with Linux support because it completely support virtual spaces on MMU (memory management unit). It delivers the most powerful debugging features by providing hardware debugging thanks to the powerful JTAG ICE. You will enjoy an experience almost like working on ITRON or OS-less environments. PARTNER-Jet enable debug the Linux target without modifications to its kernel and applications. You can also enjoy more advanced debug features by applying some simple modifications to your kernel and/or applications. The PARTNER-Jet supports leading-edge technology such as multi-core debugging, and plan to support SMP (symmetric multi-processor) core. If you are thinking about incorporating these technology, PARTNER-Jet is an ideal solution.



PARTNER-Jet Linux Support

Kernel Version: kernel 2.4, 2.6 series

Distribution: MontaVista PRO, CEE, PREVIEW-KIT, Lineo uLinux, TimeSys Linux, axLinux and more

Supported CPU: ARM, MIPS, SH which are supported by PARTNER-Jet

Linux support package (software and documents) is needed for Linux debug. Please contact for more detailed information about Linux support package.

Kyoto Microcomputer Co., Ltd.

<http://www.kmckk.co.jp>
e-mail : en-info@kmckk.co.jp



Linux programs and their characteristic viewed from the debugger

Types of Programs	Memory Space	Address	Paging	Support for Debugging
Boot loader	Non MMU space	Fixed address	Nothing	Same as ordinary embedded debugging
Kernel	Non multiplexed kernel space on MMU	Fixed address	Almost nothing	Almost same as ordinary embedded debugging
Loadbale module	Non multiplexed kernel space on MMU	Relocatable	Ondemand paging	Requires resolving relocation and support for paging
Application	Logical mutiplexed virtual space on MMU	Fixed address	Ondemand paging	Requires logical multiplexed virtual space, and support for paging
Shared library	Logical multiplexed virtual space on MMU	Relocatable	Ondemand paging	Requires resolving relocation, logical multiplexed virtual space, and support for paging

As shown above, debugging the Linux using ICE requires to support ondemand paging, relocatable format, and virtual space using MMU. The PARTNER-Jet supports all these requirements, and it is the perfect debug environment for Linux. It has been believed that it is very difficult for ICE to support logically multiplexed virtual spaces where the Linux applications are running. KMC has been working on this challenge and the PARTNER-Jet successfully cleared it.

Comparison Linux debug features

		PARTNER-Jet	Common ICE	gdb(ptrace)	kgdb
Debug target	Kernel			x	
	Module		3	x	
	Application		x		x
	Special application(init=processID1)		x	x	x
	XIP application 1		x	2	x
	Multi-process		x		x
	Multi-thread		x		x
Debug feature (Kernel space)	Software break				
	Hardware break(Command Execution)				6
	Hardware break(Data Access)				6
	Execution trace(Branch Trace)		4		x
	Source level debug				
	Auto relocation of loadable module		5		Uninvestigated
	Resolve ondemand paging		x		
	Process debug during kernel break		x		x
Debug feature (User space)	Software break				
	Hardware break(Command Execution)			6	
	Hardware break(Data access)			6	
	Execution trace(Branch Trace)			x	
	Source level debug				
	Resolve ondemand paging				
	Auto relocation of shared library				
	Attach to executing process				
	Kernel debug during process break			x	
	Execution kernel during process break and other process	7			

- 1 XIP application is a type of application which text section of application laid out into ROM devices such as flash memory.
- 2 Possible to debug gdb, if application debug feature of XIP is implemented into the kernel.
- 3 Need to implement relocation feature on ICE. Unable to debug the part of page out.
- 4 It might not be able to analyze correctly, if there is an execution trace of user space exist.
- 5 Need to have the feature to relocate the debug information.
- 6 Have to have implemented of hardware break point feature of unique CPU to the kernel debug feature.
- 7 Realizing it on virtual ICE feature.