EXHIBIT MM
(VOL 1)

Mona Baker

From: DoNotReply@ac.playstation.net
Sent: Monday, March 19, 2007 3:20 AM
To: jaymoeco@gfwireless.net
Subject: Welcome to PLAYSTATION(R)Network

Dear jason,

Welcome to PLAYSTATION(R)Network.

Your registration confirmation follows below. Please keep a copy in a secure place.

Name: jason baker
Address:
1289 16th st. NE

grand forks , ND 58201
Sign-In ID: jaymoeco@gfwireless.net
Online ID: jaymoeco
Credit Card:

Don't forget your Sign-In ID and Password. You will need it to access your account.

To change or edit registration details, sign in to your PLAYSTATION(R)Network account and select "Account Management".

If you did not intend to register for a PLAYSTATION(R)Network account, someone may have registered with your information by mistake. Contact Consumer Services for further assistance.
http://www.us.playstation.com/Corporate/ContactUs/ConsumerServices

Thank you.

The PLAYSTATION(R)Network Team

For answers to frequently asked questions and information about PLAYSTATION(R)Network terms and policies, please follow the links below.

PLAYSTATION(R)Network FAQ:
http://www.us.playstation.com/PSNfaq

Terms of Service / User Agreement and Privacy Policy:
http://www.us.playstation.com/TermsOfUse

This email has been delivered from a send-only address. Please do not reply to this message. If you have any questions, contact Consumer Services using the following link.
http://www.us.playstation.com/Corporate/ContactUs/ConsumerServices

Mona Baker

From:	DoNotReply@ac.playstation.net
Sent:	Wednesday, August 15, 2007 1:55 AM
To:	jaymoeco@gfwireless.net
Subject:	Has Been Added to Your Wallet

Dear jason,

Below is a record of the funds you added to your wallet. Be sure to keep it in a safe place for future reference.

PLAYSTATION(R)Network Wallet Transaction

Online ID jaymoeco
Name jason baker

REDACTED

Date/Time Purchased 08/14/2007 @ 11:54 PM

Funds added to wallet $10.00
Balance* $10.00

----

*This amount reflects your account balance as of the date and time of this transaction.

For answers to frequently asked questions and information about PLAYSTATION(R)Network terms and policies, please follow the links below.

PLAYSTATION(R)Network FAQ:
http://www.us.playstation.com/PSNfaq

Terms of Service / User Agreement and Privacy Policy:
http://www.us.playstation.com/TermsOfUse

This e-mail has been delivered from a send-only address. Please do not reply to this message. If you have any questions, contact Consumer Services using the following link.

http://www.us.playstation.com/Corporate/ContactUs/ConsumerServices

Mona Baker

From: DoNotReply@ac.playstation.net
Sent: Wednesday, August 15, 2007 1:55 AM
To: jaymoeco@gfwireless.net
Subject: PLAYSTATION(R)Network Purchase Confirmation

Dear jason,

Thank you for your PLAYSTATION(R)Store purchase.

A receipt of your purchase is below. Be sure to keep it in a safe place for future reference.

PLAYSTATION(R)Store Receipt

Online ID jaymoeco
Name jason baker

Order Number 228050237
Date/Time Purchased 08/14/2007 @ 11:55 PM
Total Amount $4.99

Item / Service Details Price (Dollars)

UP0017-NPUB30010_00-QBERT0000GAMEPS3-UG01 Q*bert ™ v2.0 (Full Game) $4.99

Sub Total $4.99
Tax $0.00

Total $4.99

$5.01 remains in your wallet, as of the date and time of this transaction.

For answers to frequently asked questions and information about PLAYSTATION(R)Network terms and policies, please follow the links below.

PLAYSTATION(R)Network FAQ:
http://www.us.playstation.com/PSNfaq

Terms of Service / User Agreement and Privacy Policy:
http://www.us.playstation.com/TermsOfUse

This e-mail has been delivered from a send-only address. Please do not reply to this message. If you have any questions, contact Consumer Services using the following link.
http://www.us.playstation.com/Corporate/ContactUs/ConsumerServices

Mona Baker

From: DoNotReply@ac.playstation.net
Sent: Wednesday, August 15, 2007 2:27 AM
To: jaymoeco@gfwireless.net
Subject: PLAYSTATION(R)Network Purchase Confirmation

Dear jason,

Thank you for your PLAYSTATION(R)Store purchase.

A receipt of your purchase is below. Be sure to keep it in a safe place for future reference.

PLAYSTATION(R)Store Receipt

Online ID jaymoeco
Name jason baker

Order Number 228051052
Date/Time Purchased 08/15/2007 @ 12:26 AM
Total Amount $2.99

Item / Service Details Price (Dollars)

UP9000-NPUA80063_00-VER02180407PACK1-UG02   Super Rub a Dub™ (Full Game) $2.99

Sub Total $2.99
Tax $0.00

Total $2.99

$2.02 remains in your wallet, as of the date and time of this transaction.

For answers to frequently asked questions and information about PLAYSTATION(R)Network terms and policies, please follow the links below.

PLAYSTATION(R)Network FAQ:
http://www.us.playstation.com/PSNfaq

Terms of Service / User Agreement and Privacy Policy:
http://www.us.playstation.com/TermsOfUse

This e-mail has been delivered from a send-only address. Please do not reply to this message. If you have any questions, contact Consumer Services using the following link.
http://www.us.playstation.com/Corporate/ContactUs/ConsumerServices

Mona Baker

From: no-reply@playstation.sony.com
Sent: Saturday, January 12, 2008 4:52 PM
To: jaymoeco@gfwireless.net
Subject: PlayStation(R) Information You Requested

Do not respond to this system-generated email. You will not receive a response.

Dear PLAYSTATION(R)3 computer entertainment system owner,

Sony Computer Entertainment America, Inc. (SCEA) would like to apologize for any inconvenience you experienced with your PLAYSTATION 3 system.

In order to ensure proper handling and processing, you must complete the Consumer Service Product Service Guide, which can be found at the link below. Once you fill out the form, you will need to insert the top page along with your PlayStation product before you mail it to us.

When you fill out the form, please note that you must also include the following Service Request Number:

Service Request Number: 1-141115782

Please verify the information below is accurate and correct:

Name: JASON BAKER
Phone number:
Address Line 1:
Address Line 2:
City, State/PR, Zip/Postal Code, Country: , ,

Please note: SCEA will be sending a pre-paid shipping box to the address listed above. This address will also be used for the return shipping of your unit.

Please contact SCEA Consumer Service at 1-800-345-7669 for any questions with the above information.

To proceed and print out the Service Guide, please click on this link:

http://www.us.playstation.com/whattosend/ps3/default.html

(If clicking the link does not work, please copy and paste the entire link into your Web browser.)

We appreciate your cooperation.

SCEA Consumer Services

Additional Notes:

If the link above does not work, or if you need assistance with downloading the form, please contact us via email. You can email us at:
http://www.us.playstation.com/Corporate/ContactUs/ConsumerServices

Mona Baker

From: PlayStation Consumer Services [PlayStation_Consumer_Services@playstation.sony.com]
Sent: Tuesday, February 05, 2008 6:00 PM
To: jaymoeco@gfwireless.com
Subject: Service Survey from Sony PlayStation

******Please do not reply to this email. You will not receive a response.******

Sony Computer Entertainment America: Consumer and Product Service Survey

Thank you for your recent contact to Sony Computer Entertainment America.

We continually strive to provide you with high quality support and service and your comments and suggestions enable us to better serve your needs.

Please take a moment to let us know what you think of the service you recently received by completing the attached brief survey. Please Note: You will need your 9-11 digit Service Request number which is located on your Service Receipt (ex. SR Number: 1-xxxxxxxxxx), or your phone number, with area code, that you provided when you called Consumer Services. We will also need to know what type of product you had exchanged/repaired (e.g. PlayStation®2 computer entertainment system, PSP® portable entertainment system or PLAYSTATION®3 computer entertainment system).

If you did not have your PlayStation product serviced, or if you feel you received this email in error, please email us by clicking here.

Should you need to reach our Consumer Services Department, please contact us at:
Phone: 1-800-345-7669
Hours of Operation
Monday through Saturday 6:00 A.M – 8:00 P.M
Sunday 7:00 A.M. – 6:30 P.M. Pacific Time

We appreciate you taking the time to provide your feedback.

Consumer Services

Sony Computer Entertainment America

Click Here

Please note that you can opt out of future surveys by clicking the "opt out" link below. To view our privacy policy, please click here.

Click here to be removed from our survey recipient list.

Mona Baker

From: DoNotReply@ac.playstation.net
Sent: Thursday, April 03, 2008 10:23 PM
To: jaymoeco@gfwireless.com
Subject: Funds have been added to your wallet

Dear jason,

Below is a record of the funds you added to your wallet. Be sure to keep it in a safe place for future reference.

PLAYSTATION(R)Network Wallet Transaction

Online ID jaymoeco
Name jason baker

REDACTED

Date/Time Purchased 04/03/2008 @ 08:22 PM

Funds added to wallet $5.48
Balance* $6.25

----

*This amount reflects your account balance as of the date and time of this transaction.

For answers to frequently asked questions and information about PLAYSTATION(R)Network terms and policies, please follow the links below.

PLAYSTATION(R)Network FAQ:
http://www.us.playstation.com/PSNfaq

Terms of Service / User Agreement and Privacy Policy:
http://www.us.playstation.com/TermsOfUse

This e-mail has been delivered from a send-only address. Please do not reply to this message. If you have any questions, contact Consumer Services using the following link.

http://www.us.playstation.com/Corporate/ContactUs/ConsumerServices

Mona Baker

From: DoNotReply@ac.playstation.net
Sent: Thursday, April 03, 2008 10:23 PM
To: jaymoeco@gfwireless.com
Subject: PLAYSTATION(R)Network Purchase Confirmation

Dear jason,

Thank you for your PLAYSTATION(R)Store purchase.

A receipt of your purchase is below. Be sure to keep it in a safe place for future reference.

PLAYSTATION(R)Store Receipt

Online ID jaymoeco
Name jason baker

Order Number 310594145
Date/Time Purchased 04/03/2008 @ 08:22 PM
Total Amount $6.25

Item / Service Details Price (Dollars)

UP0002-BLUS30074_00-NODOUBTTRACKPACK-UA01   No Doubt Track Pack (Add-On Content) $6.25

Sub Total $6.25
Tax $0.00

Total $6.25

$0.00 remains in your wallet, as of the date and time of this transaction.

For answers to frequently asked questions and information about PLAYSTATION(R)Network terms and policies, please follow the links below.

PLAYSTATION(R)Network FAQ:
http://www.us.playstation.com/PSNfaq

Terms of Service / User Agreement and Privacy Policy:
http://www.us.playstation.com/TermsOfUse

This e-mail has been delivered from a send-only address. Please do not reply to this message. If you have any questions, contact Consumer Services using the following link.
http://www.us.playstation.com/Corporate/ContactUs/ConsumerServices

Mona Baker

From: DoNotReply@ac.playstation.net
Sent: Tuesday, September 23, 2008 11:38 PM
To: jaymoeco@gfwireless.com
Subject: PLAYSTATION(R)Network Purchase Confirmation

Dear jason,

Thank you for your PLAYSTATION(R)Store purchase.

A receipt of your purchase is below. Be sure to keep it in a safe place for future reference.

PLAYSTATION(R)Store Receipt

Online ID jaymoeco
Name jason baker

Order Number 902704112
Date/Time Purchased 09/23/2008 @ 09:38 PM
Total Amount $22.35

Item / Service Details Price (Dollars)

EP9000-BCES00011_00-100G3T0000000000-U001	She Drives Me Crazy - Fine Young Cannibals (SingStar) $1.49
EP9000-BCES00011_00-100LSU0000000000-U001	River of Dreams - Billy Joel (SingStar) $1.49
EP9000-BCES00011_00-100JRX0000000000-U001	Opposites Attract - Paula Abdul (SingStar) $1.49
EP9000-BCES00011_00-100K6X0000000000-U001	Only The Lonely (Know The Way I Feel) - Roy Orbison (SingStar) $1.49
EP9000-BCES00011_00-10000D0000000000-U001	Oh, Pretty Woman - Roy Orbison (SingStar) $1.49
EP9000-BCES00011_00-10000J0000000000-U001	Take On Me - A-ha (SingStar) $1.49
EP9000-BCES00011_00-1000120000000000-U001	Complicated - Avril Lavigne (SingStar) $1.49
EP9000-BCES00011_00-100GXE0000000000-U001	Sk8er Boi - Avril Lavigne (SingStar) $1.49
EP9000-BCES00011_00-100K6R0000000000-U001	I Go To Extremes - Billy Joel (SingStar) $1.49
EP9000-BCES00011_00-1000H20000000000-U001	Uptown Girl - Billy Joel (SingStar) $1.49
EP9000-BCES00011_00-100K740000000000-U001	Just The Way You Are - Billy Joel (SingStar) $1.49
EP9000-BCES00011_00-100K730000000000-U001	Piano Man - Billy Joel (SingStar) $1.49
EP9000-BCES00011_00-100HUS0000000000-U001	Everybody Hurts - R.E.M. (SingStar) $1.49
EP9000-BCES00011_00-100ICI0000000000-U001	Stand - R.E.M. (SingStar) $1.49
EP9000-BCES00011_00-100GXL0000000000-U001	Should I Stay Or Should I Go - The Clash (SingStar) $1.49

Sub Total $22.35
Tax $0.00

Total $22.35

$0.00 remains in your wallet, as of the date and time of this transaction.

For answers to frequently asked questions and information about PLAYSTATION(R)Network terms and policies, please follow the links below.

PLAYSTATION(R)Network FAQ:
http://www.us.playstation.com/PSNfaq

Terms of Service / User Agreement and Privacy Policy:
http://www.us.playstation.com/TermsOfUse

This e-mail has been delivered from a send-only address. Please do not reply to this message. If you have any questions, contact Consumer Services using the following link.
http://www.us.playstation.com/Corporate/ContactUs/ConsumerServices

Mona Baker

From: DoNotReply@ac.playstation.net
Sent: Tuesday, September 23, 2008 11:38 PM
To: jaymoeco@gfwireless.com
Subject: Funds have been added to your wallet

Dear jason,

Below is a record of the funds you added to your wallet. Be sure to keep it in a safe place for future reference.

PLAYSTATION(R)Network Wallet Transaction

Online ID jaymoeco
Name jason baker

Order Number 902704105
Transaction Date 09/23/2008 @ 09:38 PM

REDACTED

Funds added to wallet $22.35
Balance* $22.35

---

*This amount reflects your account balance as of the date and time of this transaction.

For answers to frequently asked questions and information about PLAYSTATION(R)Network terms and policies, please follow the links below.

PLAYSTATION(R)Network FAQ:
http://www.us.playstation.com/PSNfaq

Terms of Service / User Agreement and Privacy Policy:
http://www.us.playstation.com/TermsOfUse

This e-mail has been delivered from a send-only address. Please do not reply to this message. If you have any questions, contact Consumer Services using the following link.

http://www.us.playstation.com/Corporate/ContactUs/ConsumerServices

From:	DoNotReply@ac.playstation.net
Sent:	Saturday, December 20, 2008 12:32 PM
To:	jaymoeco@gfwireless.com
Subject:	PlayStation(R)Network Purchase Confirmation

Dear jason,

Thank you for your PlayStation(R)Store purchase.

A receipt of your purchase is below. Be sure to keep it in a safe place for future reference.

PlayStation(R)Store Receipt

Online ID jaymoeco
Name jason baker

Order Number 966595880
Date/Time Purchased 12/20/2008 @ 10:32 AM
Total Amount $14.90

Item / Service Details Price (Dollars)

UP0101-BLUS30112_00-SONGWEAREFAM0000-UA02 We Are Family - (We Are Family) $1.49

UP0101-BLUS30112_00-SONGCELEBRAT0000-UA02 Celebration - (Celebration) $1.49

UP0101-BLUS30112_00-SONGPROUDMARY000-UA02 Proud Mary - (Proud Mary) $1.49

UP0101-BLUS30112_00-SONGSWEETHOME000-UA02 Sweet Home Alabama - (Sweet Home Alabama) $1.49

UP0101-BLUS30112_00-SONGTIMEAFTER000-UA02 Time After Time - (Time After Time) $1.49

UP0101-BLUS30112_00-SONGALWAYSSOMETH-UA02 Always Something There To Remind Me - (Always Something There To Remind Me) $1.49

UP0101-BLUS30112_00-SONGJESSIES00000-UA02 Jessie's Girl - (Jessie's Girl) $1.49

UP0101-BLUS30112_00-SONGWHOCAN000000-UA02 Who Can It Be Now? - (Who Can It Be Now?) $1.49

UP0101-BLUS30112_00-SONGCALIFORNIA00-UA02 California Dreamin' - (California Dreamin') $1.49

UP0101-BLUS30112_00-SONGPLEASEMR0000-UA02 Please Mr. Postman - (Please Mr. Postman) $1.49

Sub Total $14.90
Tax $0.00

Total $14.90

$0.00 remains in your wallet, as of the date and time of this transaction.

For answers to frequently asked questions and information about PlayStation(R)Network terms and policies, please follow the links below.

PlayStation(R)Network FAQ:
http://www.us.playstation.com/PSNfaq

Terms of Service / User Agreement and Privacy Policy:
http://www.us.playstation.com/TermsOfUse

This e-mail has been delivered from a send-only address. Please do not reply to this message. If you have any questions, contact Consumer Services using the following link.
http://www.us.playstation.com/Corporate/ContactUs/ConsumerServices

Mona Baker

From: DoNotReply@ac.playstation.net
Sent: Saturday, December 20, 2008 12:32 PM
To: jaymoeco@gfwireless.com
Subject: Funds have been added to your wallet

Dear jason,

Below is a record of the funds you added to your wallet. Be sure to keep it in a safe place for future reference.

PlayStation(R)Network Wallet Transaction

Online ID jaymoeco
Name jason baker

Order Number 966595854
Transaction Date 12/20/2008 @ 10:31 AM

REDACTED

Funds added to wallet $14.90
Balance* $14.90

---

*This amount reflects your account balance as of the date and time of this transaction.

For answers to frequently asked questions and information about PlayStation(R)Network terms and policies, please follow the links below.

PlayStation(R)Network FAQ:
http://www.us.playstation.com/PSNfaq

Terms of Service / User Agreement and Privacy Policy:
http://www.us.playstation.com/TermsOfUse

This e-mail has been delivered from a send-only address. Please do not reply to this message. If you have any questions, contact Consumer Services using the following link.

http://www.us.playstation.com/Corporate/ContactUs/ConsumerServices

SoC drawer: The Cell Broadband Engine chip: High-speed offload for the masses

Leading the broadband revolution and helping save the planet

Sam Siewert (Sam.Siewert@Colorado.edu), Adjunct Professor, University of Colorado

Summary: Cell Broadband Engine™ (Cell/B.E.) chips are leading the broadband revolution in computing and provide the core silicon DNA for supercomputing, medical image processing, and many emergent applications, as worldwide connectivity and bandwidth continue to change the world we live in. This article explores the performance of application code on the Sony® PLAYSTATION 3®'s Cell Broadband Engine system running Yellow Dog Linux®. A simple program demonstrates how multithreaded applications that use the Synergistic Processing Elements to offload work can enjoy tremendous speedup.

Date: 17 Apr 2007
Level: Introductory
Also available in: Japanese

Activity: 5619 views
Comments: 0 (Add comments)

★★★★★ Average rating (based on 8 votes)

This article provides an overview of installing and using Yellow Dog Linux on the Sony PLAYSTATION 3 (PS3) to explore the capabilities of the Cell/B.E. processor. The PS3 provides an amazing low-cost platform for multithreaded data- and compute-intensive applications. It's very accessible and easy to program, and fun, too.

In this article, I'll use the Cell/B.E. SDK to build a benchmarking application using POSIX threads that are mapped onto the processor's Synergistic Processing Elements (SPEs) over the Element Interconnect Bus (EIB). (See the Resources section below for a link to the SDK.) The Cell/B.E. processor is an advanced SoC design that provides 205 GFLOPS of performance running at 3.2GHz with an symmetric multithreaded Power Processing Element (PPE) and up to eight SPEs that can be used for offloading work. On the PS3, one SPE is dedicated to the built-in Sony GameOS, also known as the hypervisor, and one SPE is disabled to increase PS3 yield and lower cost. So, on the Yellow Dog Linux PS3 platform used in this article, the PPE has six SPEs to which it can offload over the EIB.

I'll walk you through some Pthreads-based code that you can download and use to compare Cell/B.E. processor performance to most other multicore or symmetric multithreaded architectures. I think you'll find that, compared to most other options, the Cell/B.E. processor provides truly amazing offload performance at low cost and power -- especially on the PS3.

Finally, I'll discuss the capabilities that the Cell/B.E. processor brings to systems and its potential for use in embedded or large, scalable clustered systems. The amazing capability of the Cell/B.E. processor will undoubtedly revolutionize many emergent applications in broadband, graphics, and high-performance computing (HPC), and is leading the way in SoCs joining mainstream computing.

A few notes on getting your PS3 system going

It is easy to get Yellow Dog Linux running on the PS3; however, I would like to point you to the resources I used and note a few insights that helped make my install painless and fun. Also, please note that the developerWorks Power Architecture technology zone includes many great PS3 Linux articles that helped me as well. (See the Resources section for links.)

- **Yellow Dog Linux (YDL):** This Linux (2.6.16 kernel) distribution for Power Architecture™ systems from Terra Soft Solutions makes running Linux on PS3 simple and includes a great guide for installation on the PS3. (See Resources.)
- **An HDMI-to-DVI-D or component cable for HDTV:** YDL and the Enlightenment X Window manager is best enjoyed at 1080p (1080 progressive scan resolution) on an HDTV. If you have an HD monitor or TV, I highly recommend getting the right HDMI, component, or HDMI-to-DVI cable so you can work in a windowed environment. (See Resources.)
- **Install from DVD and use a thumb drive:** The bootloader installer and bootloader (kboot) are most easily installed from a thumb drive that you can plug into one of the many USB 2.0 ports on the PS3. Likewise, YDL can be purchased or downloaded and burned onto a single-layer DVD for simple Linux install. I found the thumb drive to be a great way to transfer code as well, because the wireless 802.11g interface is not yet available in YDL and I don't have Gigabit Ethernet down by my HDTV (yet). I was able to simply pop in the thumb drive in YDL and mount it with the following command:

  ```
  mount -t msdos /dev/sdf1 /mnt/thumb
  ```

- **Don't worry about GameOS:** When I was installing Linux, it was clear that the PS3 was designed to accommodate the installation of a second OS, and offers protection mechanisms to prevent installation glitches from harming GameOS. GameOS is really a hypervisor that also manages the second OS boot and install and provides fail-safe options along with kboot to restore your PS3 back to its shipped configuration. The 60GB PS3 provides up to 10GB for GameOS and almost 50GB for Linux (or vice versa).

Jumping into SPE offloading enthusiastically

Yellow Dog Linux runs on many Power Architecture platforms

YDL runs on most Power Architecture platforms, including Apple G3, G4, and G5 machines; the Sony PS3; and embedded and HPC systems from IBM and Mercury. For example, Mercury has a 1RU dual Cell/B.E. system with dual Gigabit Ethernet and PCI-e expansion slots. At the time this article was written, YDL includes Version 2.6.16 of the Linux kernel, which corresponds to Fedora Core 5 and SUSE 10.0; it will undoubtedly be upgraded soon to the 2.6.18 kernel to match the recently released Fedora Core 6 and SUSE 10.2. For those interested in the embedded and HPC aspects of Cell/B.E. systems, this means that you can easily develop code at home on a low-cost, easy-to-use Linux platform that is instruction-set-architecture compatible with many of the new and exciting Cell/B.E. HPC and embedded systems.

The whole reason I was interested in getting a PS3 to run Linux was to see just how well the SPE offload in a Cell/B.E. system worked. Okay, I wanted to justify HD-quality Madden Football, too, but I'll guess that the SPE offload is what you're most interested in. Playing a game or two on the PS3, it's clear that there's some real compute power under the hood, but writing your own code is truly believing.

The C code found in Listing 1 includes three basic benchmarks, using iterations of a 64-bit Fibonacci sequence:

- **SPE offloading of threads:** One to six threads are created at a time so that work is offloaded to all available SPEs by the `testThread()` function. Upon completion of a set of six threads, the next set of SPE offload threads are begun. This continues until all threads have been executed on the SPEs. The Linux `gettimeofday()` function is called to time this segment of code, including completion time for all SPE threads started.
- **POSIX Pthread threads:** This segment follows the SPE thread test and creates a POSIX thread for each Fibonacci series calculation working in a loop until all are created and active, and then waits in a loop for all to finish. Again, `gettimeofday()` is called to time the segment.
- **Sequential iteration:** The Fibonacci sequence code is called iteratively multiple times to match the number of workers that were executed in the SPE and Pthread tests above.

Find the full source for this code in the Download section of this article. The code and makefile found in src1.zip is the first version of the code. It was written using POSIX threads only; src2.zip contains a revised version of the code that includes SPE offload. The exact code run, including the makefile, is provided. To build the PS3 SPE offload version, use `make spetest`; to build the simple Pthreads version, use `make`. The Pthreads and sequential test will run on any Linux platform, but the SPE test will only run on a PS3 Linux installation.

A note on Pthreads scheduling and compiler optimization

`testThread()` itself is a Pthread that runs using first-in, first-out (FIFO) policy at rtmax-1 so that it won't be interfered with during the testing. All Pthreads created for the test are run FIFO policy at rtmax priority so that they won't be interfered with at all other than by critical system interrupts. This careful Pthread scheduling leads to much more deterministic and repeatable test runs. Furthermore, the workload code implementing the Fibonacci sequence has been carefully designed so that it can not be over-optimized and will yield a repeatable workload for each threading method tested.

Listing 1. Pthread used to evaluate sequential, PPE threaded, and SPE threaded performance

```
void *testThread(void *threadid)
{
#ifdef SPE
   fibdata fdarray[MAX_NUM_THREADS] __attribute__((aligned(16)));
   speid_t spe_id[6];
#endif
   double DT=0.0;
   int i, j;
   int numThreads = (int)threadid;
   int threadAlloc;
```

```
#ifdef SPE

   for(i=0;i<MAX_NUM_THREADS;i++)
   {
       // Initialize data to send to each SPE
       fdarray[i].idx = 0;
       fdarray[i].jdx = 1;
       fdarray[i].seqCnt = seqIterations;
       fdarray[i].iterCnt = Iterations;
       fdarray[i].fib = 0;
       fdarray[i].fib0 = 0;
       fdarray[i].fib1 = 1;
       fdarray[i].padding = 0;
   }

   // SPE thread benchmark
   startTOD=readTOD();
   for(i=0;i<numThreads;i+=6)
   {
       if((numThreads-i) <= 6)
            threadAlloc=6;
       else
            threadAlloc=(numThreads-i);

       for(j=0;j<threadAlloc;j++)
       {
       spe_id[j] = spe_create_thread(0, &fib_spe_handle,
                                     &fdarray[i], NULL, ANY_SPE,
                                     NO_OPTIONS);
           if(spe_id[j] == 0) exit(-1);
       }
       for(j=0;j<threadAlloc;j++)
           spe_wait(spe_id[j], NULL, 0);

   }
   stopTOD=readTOD();

   DT = elapsedTOD(stopTOD, startTOD);

   printf("SPE Thread: Fib(%u)=%llu in %lf secs for Iter=%u\n",
          (seqIterations*Iterations), fdarray[0].fib, DT, fdarray[0].iterCnt);

#endif


   // Pthread benchmark
   startTOD=readTOD();
   for(i=0;i<numThreads;i++)
       pthread_create(&threads[i], &rt_sched_attr, fibSeq, (void *)i);
   for(i=0;i<numThreads;i++)
       pthread_join(threads[i], NULL);
   stopTOD=readTOD();

   DT = elapsedTOD(stopTOD, startTOD);

   printf("Threaded:   Fib(%u)=%llu in %lf secs for Iter=%u\n",
          (seqIterations*Iterations), finalFibVal[0], DT, Iterations);

   // Sequential benchmark
   startTOD=readTOD();
```

```
    for(i=0;i<numThreads;i++)
        fibSeq((void *)i);
    stopTOD=readTOD();

    DT = elapsedTOD(stopTOD, startTOD);

    printf("Sequential: Fib(%u)=%llu in %lf secs for Iter=%u\n",
           (seqIterations*Iterations), finalFibVal[0], DT, Iterations);

    }
```

The test code used to benchmark the speedup provided by the six SPEs available on the PS3 uses the Cell SDK to embed code and to pass data to the SPEs through the EIB. The `fibdata` data structure is passed down to an SPE by the PPE along with the code that is embedded through the `fib_spe_handle`. It has been well noted by the designers of the Cell/B.E. architecture that some atypical coding constructs must be used to employ the SPEs -- the method used to embed and download code along with data used in this example is the main coding paradigm that differs from typical threading. The makefile uses `spu-gcc`, an SDK compiler, to build the SPE downloadable code, which is provided in Listing 2. Furthermore, once the SPE code has been generated, it is embedded into an ELF (Executable and Linking Format) object code file and incorporated into the main program through the `fib_spe_handle`.

Following this procedure is simple, but, on the downside, the parameters passed in with the code must be 16-byte aligned; they are also not type checked or otherwise checked to ensure consistency between the two `fbdata` declarations in Listing 1 and Listing 2. Programmers should take care to make sure agreement between the types, alignment, and structure definition is correct, because any disagreement won't be caught by the compiler and will lead to a runtime error that will be harder to debug. The worst that can happen is a bus error or parameter mismatch, but the cause of such a problem may not be immediately obvious, so double-check declarations shared between PPE and SPE.

Listing 2. The Fibonacci workload program for the SPEs

```
#include "spu_mfcio.h"

typedef struct
{
    unsigned int idx;
    unsigned int jdx;
    unsigned int iterCnt;
    unsigned int seqCnt;
    unsigned long long fib;
    unsigned long long fib0;
    unsigned long long fib1;
    unsigned long long padding;
} fibdata;

int main(unsigned long long spe_id, unsigned long long fibdata_ea, unsigned long lo
{
   fibdata fd __attribute__((aligned(16)));
   int tag_id = 0;
```

```
    // Read in fibdata
    mfc_get(&fd, fibdata_ea, sizeof(fibdata), tag_id, 0, 0);
    // wait for data
    mfc_write_tag_mask(1<<tag_id);
    mfc_read_tag_status_any();

    // Compute sequence requested
    for((fd.idx)=0; (fd.idx) < (fd.iterCnt); (fd.idx)++)
    {
       fd.fib = fd.fib0 + fd.fib1;
       while((fd.jdx) < (fd.seqCnt))
       {
          fd.fib0 = fd.fib1;
          fd.fib1 = fd.fib;
          fd.fib = fd.fib0 + fd.fib1;
          (fd.jdx)++;
       }
    }

    // Write back fibdata
    mfc_put(&fd, fibdata_ea, sizeof(fibdata), tag_id, 0, 0);
    // wait to complete
    mfc_write_tag_mask(1<<tag_id);
    mfc_read_tag_status_any();

    return 0;

}
```

The code in Listing 2 is downloaded to each SPE and `fibdata` is passed in through the SDK `mfc_put()` and `mfc_get()` calls in the SPE code. Note that the SPE code is written as a new main program and code is generated for it prior to embedding through the handle into the PPE code. In this example, there is one version of the SPE code used by all SPEs, but a unique copy of `fibdata` is used for each thread instance with `fdarray`.

I first completed testing using the Pthreaded code just to take a look at the speedup provided by the PPE symmetric multithreading to the Fibonacci worker threads. If you download and run this code, the acceleration provided by the PPE SMT becomes apparent when the Fibonacci sequence is run for thousands of iterations or more. Basically, there is a point at which the overhead of thread creation and management is overcome by the speedup gained from threads being executed with SMT acceleration.

I further tested this simple Pthread code on the PPE to see how it scales with an increasing number of threads. In general, the PPE SMT provides a constant speedup that is significant. Looking at PPE SMT-based speedup as a ratio of sequential time taken divided by Pthread time taken for each thread set will show you how speedup varies with number of threads. Speedup is fairly constant as threads are scaled. The PPE itself provides significant thread scaling, but it is intended to provide control and workload management for the SPEs, which provide much greater speedup for threads.

You can best achieve huge performance advantages on Cell/B.E. systems by downloading thread code onto SPEs. You might expect that six SPEs would provide a speed improvement of about a factor of six (minus the overhead of code download and message passing), but I was pleasantly surprised to find an even greater speedup on my system. I suggest that you download the code and give it a try on the PS3 or any other Cell/B.E. system to measure speedup. I only tested the code on YDL on the PS3, but I would expect it to work on just about any Cell/B.E. platform that runs Linux. POSIX threads have been designed to be portable and the SPE offload uses the Cell/B.E. SDK.

When the SMT of the PPE and pipelining is employed using the SPEs, I found that the speedup was greater than I thought it might be on my system. The EIB allows the PPE to start downloads on multiple SPEs and to overlap their starts and stops very efficiently so that the entire process is fully pipelined. Acceleration is also provided on the SPEs by vector processing features not found on the PPE, including 128-bit wide vector processing. So, not only can the PPE efficiently start overlapping execution on all SPEs, but the code would be expected to execute faster thanks to vector processing as well. The test code provided in this article will reveal the true power of Cell/B.E. processing for threaded applications, and to some extent, the vector processing capabilities of the SPEs. Cell/B.E. processors also include AltiVec accelerating operations for graphics, image processing, and digital signal processing that will further improve performance for applications that can employ these instructions. The code provided in this article is limited to integer operations. In future articles, I plan to take a closer look at floating point workloads.

The original and true promise of broadband to our planet

The core that won back supercomputing honors

In September of 2004, the Japanese NEC Earth Simulator system was dethroned by the U.S. Blue Gene®/L system as the world's faster supercomputer. The Cell processor (closely related to Cell/B.E. chips) helped bring this honor back to the U.S. with the rollout and power on of Blue Gene/L. Blue Gene/L once again took honors on the SC06 Top 500 list with 280.6 TFLOPS.

For me, using the PS3 to run Linux brings back memories of the early days of computing, when the world was filled with the promise of silicon-based revolution. Maybe it's the programming on a TV that reminds me of the early computers. Better yet, Cell/B.E. technology has been designed to help realize the full potential of Web-based networked computing with high-end graphics and video. The World Wide Web in its early days was seen as revolutionary in that it would surely lead to work at home, less travel, less commuting, less pollution, more global communication, and a flat world with fair e-commerce, and would ultimately serve to provide more efficiency and fairness in the world as a whole -- well, at least a few idealists like myself thought this. Cell/B.E. technology is exciting because the PS3 may be the lowest-cost, highest-performance computer ever provided to the general public. Sure, the PS3 is a costly game platform, but it can do far more than play games. The idea of Cell/B.E. chips as the DNA of the fastest computer available to the masses and the core of the fastest computer period (Blue Gene/L) is truly promising. With broadband transport becoming widely available to most of the world, a multicore processor to make good use of it has now finally also been introduced.

As the world faces issues like global warming and political rifts, the emergence of new technology that can help us work more effectively at home, minimize grueling business travel, communicate better, have more fun, and get excited about computing again is a welcome sight. I have to wonder: Do current estimates of potential reductions in greenhouse gasses take into account the extent to which broadband might reduce commuting and global travel for business? Emergent new applications like telemedicine, effective high-definition video conferencing, and true virtual presence can help change the shape of things to come for the better. Either way, Cell/B.E. technology sure does help my threads run faster.

Conclusion

The PS3 is a great and relatively low-cost way to explore and evaluate the capability of Cell/B.E. technology -- and it's fun, too. It can also serve as a development platform for work on Cell-based HPC or embedded software and has the ability to serve as a great Linux platform at home. While it is short two SPEs compared to HPC/embedded Cell/B.E. platforms, it can host the same SDK and be used quite readily to develop SPE offloading code. It will be interesting to see how many PS3s wind up running Linux -- I suspect all will also be used as the game platforms they were intended to be, but the idea of a game system designed to host Linux from the beginning was in my opinion an excellent decision for both PS3 and Cell/B.E. technology. While Cell/B.E. chips and the PS3 may not solve global warming, they will keep a few people off the road and at home nights and weekends.

Downloads

Description	**Name**	**Size**	**Download method**
Sample code written using POSIX threads only	src1.zip	32KB	HTTP
Revised version of code that includes SPE offload	src2.zip	36KB	HTTP

Information about download methods

Resources

Learn

- The SoC drawer series: The Cell/B.E. chip is an SoC (system on a chip) like no other, but for more general information on SoCs, see the previous articles in this series.

- Wikipedia offers a good introduction to a number of topics covered in this article:
  - Cell/B.E. technology
  - PlayStation 3
  - HDMI
  - Component video
  - DVI

- The IBM Cell Broadband Engine resource center: Includes numerous excellent articles on Cell/B.E. chips.

- "An introduction to Linux on the PLAYSTATION 3," Jonathan Bartlett (developerWorks, January 2007): This article offers specific guidance on how to explore this cool new architecture in the comfort of your home.

- "Using advanced compiler technology to exploit the performance of the Cell Broadband Engine architecture," A. E. Eichenberger et al. (*IBM Systems Journal,* 2006: A seriously interesting application for Cell/B.E. technology from the IBM research site. (PDF format))

- Blue Gene/L: If you're still not impressed, then take a look at this Cell/B.E. technology-powered supercomputer at Lawrence Livermore National Labs.
- POSIX threading: LLNL has a great tutorial Web site on this subject.
- The Supercomputing Top 500 list: Cell/B.E. technology-based computers do well on this list.

Get products and technologies

- IBM Cell Broadband Engine Software Development Kit: Download it from alphaWorks.
- Install Manual of Open Platform for PlayStation 3: Provides a download for the bootloader used to boot and install YDL.
- 1U Dual Cell-Based System: One of the Cell/B.E. technology-based HPC and embedded solutions from Mercury Computer; this one integrates two Cell/B.E. systems into a single rack unit.
- Yellow Dog Linux: This distribution from Terrasoft Solutions made my introduction into PS3 Cell/B.E. technology-based coding totally awesome. The company's Guide to Installing YDL Linux 5.0 for PlayStation (PDF format) was easy to follow, and describes the interoperation of YDL and the PS3 hypervisor, also known as GameOS.
- More IBM resources:
    - The Cell project at IBM Research
    - Cell Broadband Engine page at IBM Semiconductor Solutions

About the author

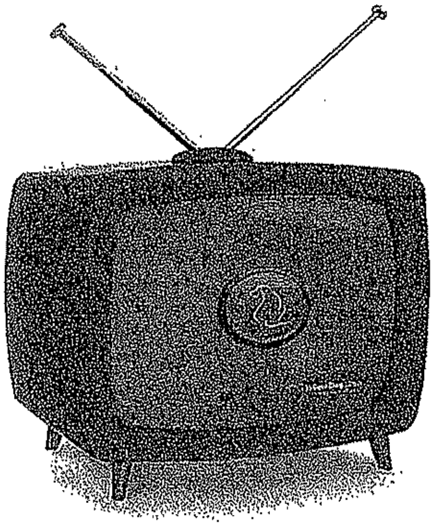
Dr. Sam Siewert is an embedded system design and firmware engineer who has worked in the aerospace, telecommunications, and storage industries. He also teaches at the University of Colorado at Boulder part-time in the Embedded Systems Certification Program, which he co-founded. His research interests include autonomic computing, firmware/hardware co-design, microprocessor/SoC architecture, and embedded real-time systems.

Trademarks | My developerWorks terms and conditions

FIXSTARS
Speed up your Business

A GUIDE TO INSTALLING

YELLOW DOG LINUX 6.2



FOR PLAYSTATION®3

Rev 1.0 June 29, 2009

A Guide to Installing Yellow Dog Linux 6.2– For PLAYSTATION®3

Legal Notices

Trademarks

PLAYSTATION3 is a registered trademark of Sony Computer Entertainment Inc.

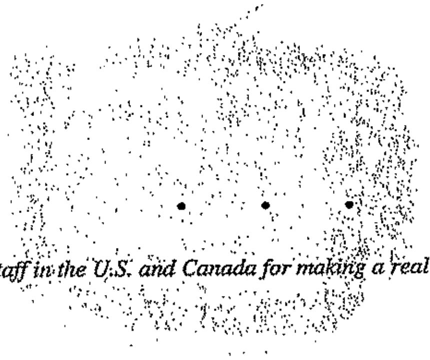
Linux is a registered trademark of Linus Torvalds.

YDL, Yellow Dog Linux, Y-HPC, and Y-Film are registered trademarks of Fixstars Solutions.

Additional product and company names mentioned may be trademarks and/or registered trademarks of their respective holders.

Copyright Notice

Fixstars Solutions
451 North Railroad Avenue, Suite 102
Loveland, CO 80537
+1 (970) 278-9243 voice
+1 (707) 988-7982 fax



Many thanks to the Fixstars staff in the U.S. and Canada for making a reality this latest release of Yellow Dog Linux, our best effort yet.

Revision History

Revision	Date	Change
1.0	June 29, 2009	Initial Publication.

INTRODUCTION

Thank you for choosing Yellow Dog Linux!

When Sony Computer Entertainment designed the PLAYSTATION®3 (PS3™), it was fully intended that you, a PS3 owner could play games, watch movies, view photos, listen to music, and run a full-featured Linux operating system that transforms your PS3 into a home computer.

Yellow Dog Linux for PS3 combines a simple to use graphical installer with leading-edge components and a foundation of must-have home, office, and server applications. Everything you need to browse the web, check and compose email, do your school homework or take your office work home is included with more than 2000 packages on the Install DVD.

This Guide to Installation

This Guide to Installation is written specifically for the Sony Computer Entertainment PLAYSTATION®3. This Guide does not offer instructions specific to other computers (ie. Apple Power Macintosh) supported by Yellow Dog Linux. Installation instructions for other computers are available from Fixstars' website.

Refer to us.fixstars.com/support/ for updates, issues specific with unique computer configurations, and engineering notes.

Where is Yellow Dog Linux Installed?

The drive installed in your PS3 ships with just one large partition, a logical division of the drive into unique sectors for the organization and protection of data.

Before you install Yellow Dog Linux, you will use the PS3 GameOS to split the drive into 2 major partitions:

- one for saving game data, music, and photos
- the other for Yellow Dog Linux

In this respect, both the GameOS and Yellow Dog Linux maintain their own unique volumes on which to store data.

Two Operating Systems on my PS3?

Following the installation of Yellow Dog Linux on your PS3, there will be 2 operating systems: the GameOS and Yellow Dog Linux (YDL).

The GameOS is the operating system that ships pre-installed on PS3. It is also referred to as a hypervisor, an operating system that provides a channel of communication between the outside world (you, your game controller, the monitor) and the built-in hardware (CPU, motherboard, RAM, hard drive, and I/O ports).

When you use the on-screen menus, play a game, watch a movie, or listen to music, you are doing so through the GameOS. Yellow Dog Linux runs on top of the GameOS which provides an important communication between Linux and the hardware.

Even if this sounds a bit technical, it is important as the GameOS provides a layer of continuity for Linux, making the development and maintenance of Linux for PS3 a more predictable process. This means you will enjoy a higher quality,higher performance Linux operating system. We hope you enjoy Yellow Dog Linux for PS3 and we look forward to receiving your feedback.

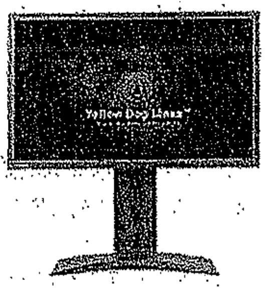
Please visit us at us.fixstars.com.

Step 1 - Attach Monitor or HD TV

Attach your PS3 to a hi-definition TV or monitor using either:

- A HDMI (digital) cable
- Component (5 RCA jacks) cable

Composite (yellow RCA jack) is ***not*** supported as the resolution is too low.



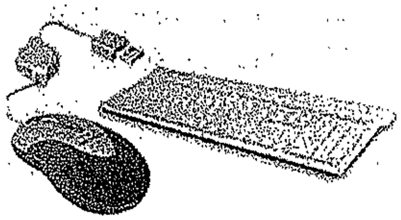
Step 2 - Attach Game Controller, Keyboard and Mouse

Attach to the PS3:

- A Game controller
- A USB keyboard
- Mouse

The game controller will be used to conduct all GameOS functions.

The USB keyboard and mouse will be used once you enter the YDL installer.



Step 3 -Power On

1. Power on your PS3.
2. If this is the first time you have used your PS3, you will be prompted to:
    - Select a language
    - Select a time zone
    - Set the time & date
    - Set the username (more easily done with a USB keyboard)

Step 4 - Update the GameOS and Backup your Data

1. It is important to update the GameOS to the most current version, available for free from Sony. To update:
   - Select **Settings > System Update > Update via Internet** and then follow the on-screen prompts to complete.
2. If you have used your PS3, it is highly recommended that you **BACK UP ALL GAME, MUSIC, & PHOTO DATA** before formatting the drive as all data will be lost.

Step 5 - Format the Hard Disk

1. Format the hard disk:
   - Select: **Settings > System Settings** and press ENTER on the keyboard or the X button on the game controller.
   - Select **Format Utility > Format Hard Disk > Yes**
2. Next, select a partition setting for the hard disk:
   - Select: **Custom**
3. You are presented with option to **Use All for the PS3 System, Allot 10GB to the Other OS** or **Allot 10GB to the PS3 System.** Allocating the majority for YDL (Other OS) is recommended, independent of the size of your drive.
4. Select **Quick Format** and **Yes.** Your PS3 will now format its internal drive.
5. Press **X** on the controller to **EXIT.** This will cause your PS3 to reboot.

Step 6 - Install the Bootloader

Before you install Yellow Dog Linux, you must temporarily transfer control of your PS3 to the Yellow Dog Linux installer. This is done using a bootloader. The bootloader installer, preloaded by Sony with the GameOS, will install the bootloader. The bootloader "kboot" allows the GameOS to transfer authority to Yellow Dog Linux. Kboot is included on the YDL Install DVD in the directory /PS3/otheros directory and called "otheros.bld".

Do not turn off the power nor remove the recording medium during installation. Interrupting the installation process may cause damage.

1. Insert YDL Install DVD.
2. From the GameOS menu, select:

   Settings > System Settings > Install Other OS
3. Your PS3 will now scan the DVD for the bootloader installer and bootloader.
4. Confirm discovery of "**/PS3/otheros/otheros.bld**".
5. Press **X** on the controller to start the installation.
6. Once the install is finished, the screen will say "Install completed." This means that the kboot install is complete.

   Note: YDL ***has not been*** installed yet; continue to the next step.
7. Back on the GameOS menu, select **Other OS** in order to begin the YDL install:

   Settings > System Settings > Default System > Other OS
8. When prompted to quit the PS3 system and start the other system, select **Yes**.

YDL INSTALLATION

When installing YDL, you will need :

- A monitor or television with HDMI (digital) or Component (5 RCA jacks) cable input in order to perform a graphical install
- A USB Keyboard

Step1 - Begin the Installation Script

1. The PS3 will boot into the kboot bootloader. You will see white text on a black screen. The system will stop at the **kboot:** prompt.

   kboot:

2. Before the installer times-out, press the **TAB** key on your keyboard to browse through the available install modes. Each time you press the **TAB** key, another install mode will be displayed. Select one of the following options to match the highest resolution capability of your monitor.

install_ppc32	Do not select this option. This is not for the PS3
install_ppc32_text	Do not select this option. This is not for the PS3
install_ppc64	Do not select this option. This is not for the PS3
install_ppc32_text	Do not select this option. This is not for the PS3
install_ps3_1080i	for PS3 and monitor capable of 1080i
install_ps3_1080p	for PS3 and monitor capable of 1080p
install_ps3_480i	same as install_text
install_ps3_720p	for PS3 and monitor capable of 720p

For example, if you monitor is capable of 1080p, the kboot prompt should look similar to this following:

kboot: install_ps3_1080p

If your monitor has only Composite (3 RCA jacks) the YDL installer Anaconda) should default to "install_text", a low-resolution text mode. This will later require command line configuration.

If you are uncertain of your hi-def (HDMI, Component) monitor's capability, the 720p option is the best default choice.

3. Once you select the appropriate install mode, hit **ENTER** on the keyboard. This will begin the installation., or Anaconda (the Yellow Dog Linux installer).

Step 2 - Language Selection

1. Select your preferred Language configuration.
2. Click **Next** to proceed.

Step 3 - Keyboard Selection

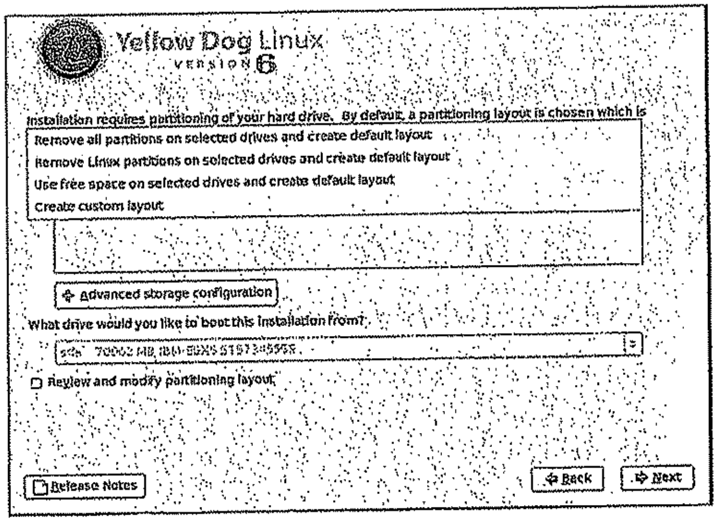
1. Select your preferred Keyboard configuration.
3. Click **Next** to proceed.

NOTE: If this is the first time installing Linux on your PS3, you will be warned, "The partition table on this device ps3da was unreadable...". This is normal. Go ahead and select "Yes" to proceed.

Step 4 - Partition Drive

Partitioning a drive creates logical (as compared to physical) sections on your drive, each serving the purpose of organizing and protecting data. You may either allow the Installer to create the partitions for you by selecting Automatic Partitioning, or conduct this effort manually via Disk Druid.

1. You may choose from one of following schemes:
   - **Remove all partitions:** on a PS3 (as compared to a Mac) this will produce the same result as "Remove all Linux partitions" (above) as the GameOS cannot be affected by the Linux installer.
   - **Remove all Linux partitions:** this will delete and then replace only existing Linux partitions. Please note that if you have already installed Linux on your PS3, this procedure will DESTROY ALL EXISTING LINUX DATA. The GameOS and game data will not be harmed.
   - **Keep all partitions and use existing free space**: if you have never installed Linux on your PS3, this will generate the same result as "Remove all partitions" (above).
   - **Custom**: manually partition using Disk Druid. This is only recommended for advanced users.



Installation requires partitioning of your hard drive. By default, a partitioning layout is chosen which is

Remove all partitions on selected drives and create default layout
Remove Linux partitions on selected drives and create default layout
Use free space on selected drives and create default layout
Create custom layout

What drive would you like to boot this installation from?

sda 70002 MB [illegible]

□ Review and modify partitioning layout

More About Custom Partitioning

The use of Disk Druid is for advanced Linux users who desire to customize the partitioning of their drive. If you are new to Linux, we recommend you use the Automatic Partition tool discussed earlier in this section.

If you elect to build your partitions manually, you should create the following:

swap - used to temporarily store data no longer held in RAM. Swap should be no less than 512MB.

home - by creating a "/home" dir, you may re-install YDL and preserve your home directory. This may be any size you desire, but likely at least 1 GB.

/ - contains everything else (/etc, /sbin, /var) and should be at least 10GB.

Step 5 - Network Configuration

1. The default DHCP enables your PS3 to automatically obtain an IP address from your DSL or cable modem or in-office router. Only advanced users who have specific needs should manually configure this information.
2. Click **Next** to proceed.

Step 6 - Time Zone Selection

1. Select the appropriate time zone based on the location of the PS3.
2. Click **Next** to proceed.

Step 7 - Root Password

1. Choose a password that is at least 8 character, both letters and numbers and is not based on a dictionary word. Lower and upper case letters are unique.

   Root Password Guidelines

   Do not use your birthday, your child's name spelled backward, nor the name of your spouse followed by the year you were married. The security of your computer is only as strong as your passwords. Do not forget this password. Only login as root if you are an experienced user, as it is possible to remove files or data which are required for the proper function of your Linux OS.

2. Click **Next** to proceed.

Step 8 - Package Selection

1. Select one of the predefined package sets:
   - Office and Productivity – For most users
   - Software Development – For software engineers
   - Web server – For system administrators

   You may elect to customize the package set to be installed, which may increase or decrease the number of applications and time required. Once up and running, additional packages may be installed using yum or the **Add/Remove Software** utility.
2. Click **Next** to proceed.

Step 9 - Dependency Check

1. The Installer will now check for package dependencies. This will take 3-4 minutes.

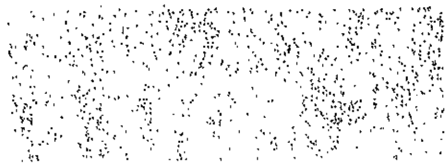
Step 10 - Installation

1. Once the dependency check is complete, click **Next** to begin the installation.
2. The install could take up to 15 minutes, depending on the number of packages you selected to install.
3. Once the install is complete, click **Reboot** to restart your PS3.
4. Remove the DVD (or it will be pulled back into your PS3 and will attempt to install again).

BOOTING UP!

If you have a low-res TV, your PS3 will require some command line modification to gain a low res (difficult to read) graphical interface. Instructions at us.fixstars.com/support/solutions/.

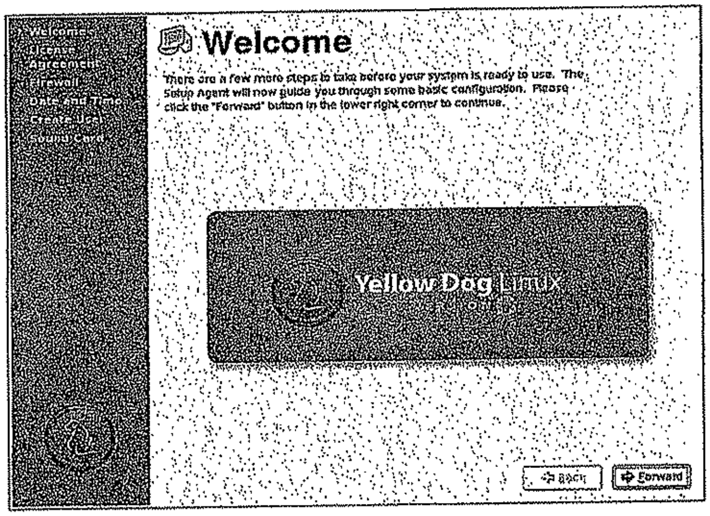
During the first-ever boot following installation, you will be welcomed by firstboot in order that you may conduct additional system tests and configuration. See the next section for details.



FIRSTBOOT

Firstboot walks you through Acceptance of the License Agreement, Date and Time settings, YDL.net Account, System User, and Sound Setup.

- **License Agreement** - Please read and accept this document.
- **Date and Time** - If your PS3 will always be connected to the internet, selecting NTP will make certain your clock is accurate. If not, manually set the Date and Time.
- **YDL.net Username and Password** - If you have purchased a YDL.net enhanced account, you have the option to enable the system to download yum updates from the YDL.net repositories instead of from the public mirrors. To enable this feature, select the YDL.net checkbox and enter your YDL.net username and password.
- **System User** - Create one user account for your initial login and immediate access to the system. Additional users may be added later. Choose a password that is at least 8 characters, both letters and numbers and is not based on a dictionary word. Do not use your birthday, your child's name spelled backward, nor the name of your spouse followed by the year you were married. The security of your computer is only as strong as your passwords.
- **Sound Setup** - You must have audio cables attached to your monitor/TV in order to hear the output. Press the play button and confirm that music was heard.



Welcome

There are a few more steps to take before your system is ready to use. The Setup Agent will now guide you through some basic configuration. Please click the "Forward" button in the lower right corner to continue.

Logging In

At the graphical prompt, enter the username and password created during firstboot's System User query. You will then enter the graphical user interface to Yellow Dog Linux, featuring the Enlightenment "E17" desktop environment.

With Linux, no one desktop environment is locked to the operating system, granting end users room for creativity, choice, and a variety of features.

Other desktop environments such as KDE and Gnome are available with the YDL Install DVD or from on-line repositories. KDE offers a tightly integrated productive environment for office and education. Gnome generally caters to those who desire greater room for customization. Gnome is installed by default and may be selected at log in.

Yellow Dog Linux defaults to the Enlightenment "E17" desktop environment.

Enlightenment

Enlightenment "E17" is an alternative desktop manager to KDE and Gnome, offering a completely customizable interface, integrated functionality, access to all the traditional applications, and a lightweight, intuitive, yet robust graphical interface.

While unique in many respects, E17 offers familiar operating system features such as a start (YDL logo) menu, productivity applications, and the ability to customize the desktop environment to your liking.

To learn more about Yellow Dog Linux and the E17 desktop environment, review the included Quick Start Tutorial, presented as the home page of your FireFox web browser:

YDL Menu > Applications > Internet > Firefox

The same tutorial is also available on-line:

us.fixstars.com/support/configuration/

Returning to the GameOS

Your PS3 will now boot to YDL each time it powers on unless you command kboot (the bootloader) to again boot the GameOS (see BOOTING GAMEOS later in this guide). To return to the GameOS, at the kboot prompt, enter:

```
ps3-boot-game-os [ENTER]
```

The PS3 may also be forced to return to the GameOS by holding the power button for about 5 seconds at power on. Doing this resets your PS3 to factory default.

Returning to YDL

To return to Yellow Dog Linux:

1. Select: **Settings > System Settings > Default System > Other OS**
2. When prompted to boot Other OS, select **Yes.**

SUPPORT

Installation Support

Installation Support is provided through YDL.net Enhanced accounts. To purchase a YDL.net Enhanced account, visit:

us.fixstars.com/store/

... and select the "Yellow Dog Linux" icon. Once purchased, you may proceed to www.ydl.net create your new YDL.net Enhanced account. Once logged in, follow the Help links to Support on the left side bar.

HOWTOs, Lists, and Forums

A guide to free, on-line support is available at Fixstars' website:

us.fixstars.com/support/

Technical Support

To learn more about Fixstars' technical support offerings, visit:

us.fixstars.com/support/

Linux development on the PlayStation 3, Part 1: More than a toy

Peter Seebach, Freelance author, Plethora.net

Summary: The Sony PlayStation 3 (PS3) runs Linux®, but getting it to run well requires some tweaking. In this article, first in a series, Peter Seebach introduces the features and benefits of PS3 Linux, and explains some of the issues that might benefit from a bit of tweaking.

Date: 18 Mar 2008
Level: Intermediate
PDF: A4 and Letter (35KB | 8 pages)Get Adobe® Reader®
Also available in: Russian Japanese Portuguese

Activity: 115601 views
Comments: 4 (View or add comments)

★★★★★ Average rating (based on 42 votes)

- **Show articles and other content related to my search: playstation 3**

When Sony first announced that the PlayStation 3 would be able to run Linux natively, a great deal of excitement ensued. Early on, it was a bit of a challenge to get Linux natively installed. The supported installer ran a custom script that hand-mangled a Fedora Core 5 or 6 install DVD into a runnable system with a special PS3 kernel. People put in hours and hours of effort to get other systems, such as Ubuntu, working. Terrasoft's Yellow Dog Linux, with an actual graphical installer that ran on the PS3, was the king of the hill.

Time has passed, and a great deal has changed. Fedora 7 installs on the PS3 out of the box, with the most challenging parts being selecting the PS3 storage driver so it can find the install DVD, and remembering to specify a video mode on the initial install command line so you'll be able to see the installer.

So, you can run Linux on the PS3. It's easy. The problem is, it doesn't necessarily run well. If you picked the PS3 up as a cheap Cell development system, it's a little frustrating to discover that, having followed the default install procedure, you have a system that comes up with a hundred megs or more of swap in use by the time you get to a shell prompt.

About this series

This series of three articles looks at PS3 Linux as a prospective development environment.

This first article, Part 1, introduces the basic configuration knobs and widgets specific to the PS3, shows you how to use them effectively, and suggests the kind of trickery that might get improved performance or a more usable display.

Part 2 and Part 3 then delve into some of the performance and tuning issues that, while they might apply on any system, are particularly useful for turning your PS3 from a proof-of-concept demo into a working system.

Getting started

Grab Fedora 7. (Fedora 8 probably works too, but I had Fedora 7 handy when I started typing.) Obviously, you might get better results in some ways from one of the more natively tuned distributions, or one of the distributions built around being small and efficient, but Fedora 7 has a huge advantage: it's supported by the Cell/B.E. SDK. (So is RHEL 5.1, but Fedora 7's nicer.)

I'm a big fan of using Bittorrent for downloads, because it helps spread the load out. This means that you should leave your client running until you've uploaded at least as much as you've downloaded; open source software is about cooperation, and cooperating on bandwidth costs is a great starting point. It'll take a while to download the whole DVD image (see Resources), so while you're waiting, here's some background on PS3 Linux.

The PlayStation 2 ran Linux as well, but it was a bit of a pain. Sony released a special kit for it, which ran only on the older, thick PS2s and required an optional hard drive. There was no support for the later (more common) "slim" PS2 units, and the kit quickly disappeared. But, fundamentally, it disappeared because it was mostly a proof of concept; the system didn't have all that much memory, and without special compiler tools, the CPU was just a fairly standard MIPS core. The vector units were not widely supported or documented outside of proprietary tools.

By contrast, the PS3 has had support for running Linux since before it was released, and Sony has made support code and packages widely available. The "Cell addons" CD released around the system's launch included direct support for Fedora 5 and 6, but it also provided all the files needed for other distributions to develop PS3 ports. As a result, Fedora 7 simply works on the PS3 without the need for a separate CD, although you still need to use the Sony downloads to install the bootloader. The add-on software CD (see Resources) continues to get updates with new kernel versions.

Understanding the hypervisor

The PS3's base specs are 256MB of main memory, a Cell/B.E. processor, wired ethernet (and 802.11b/g wireless, on some models), a graphics unit called RSX (with 256MB of its own memory), and a SATA disk drive. There are also some peripheral attachments available, including USB ports, a card reader on some models, and Bluetooth. However, this is not the machine PS3 Linux runs on.

Where's the other SPE?

The Cell Broadband Engine has 8 SPEs. PS3 Linux has 6. One is reserved by the hypervisor. Where's the other one?

The answer is that the PS3 ships with one SPE disabled. This allows the use of chips where a single SPE was defective, improving chip yields and reducing costs. So, although the theoretical design of the Cell/B.E. has 8 SPE cores, there are only 7 available on PS3 hardware. One of them runs only security tasks assigned to it by the hypervisor, leaving 6 available for Linux, or for games.

On the PS3, Linux runs under a hypervisor—in essence, PS3 Linux is running on a virtualized machine, similar to a Xen virtualized machine. The hypervisor provides controlled and filtered access to the PS3's hardware. For instance, Linux code has no direct access to the RSX graphics processor.

Rather, a special device is provided that allows limited access to it, providing for a reasonably efficient 2D framebuffer.

The hypervisor provides a number of translations. For instance, as mentioned, the hypervisor provides Linux with a virtual framebuffer device, which the hypervisor then implements using the RSX graphics core. Similarly, Linux has no access to the SATA hard drive. Instead, it has access to a SCSI drive, which maps only onto the region of the internal drive that was allocated for the "Other OS" partition. It is simply physically impossible to access the part of the disk being used by PS3 game software.

The network hardware is perhaps the most interesting. PS3 Linux can access either the gigabit wired ethernet or the 802.11g wireless, but not both at once. When you configure the network interface, if you specify it as a wireless interface with an SSID. The hypervisor uses the wireless; otherwise, it uses the wired interface.

Understanding consumer electronics video modes

If you're reading this with a PC hardware background, the video modes are going to sound weird. Rather than describing them in terms of total resolution ("1024x768") or given cryptic names ("XGA", "SXGA", and so on), most of the video modes used are described exclusively in terms of vertical lines of resolution. In the US, using the NTSC standard, the usual resolution of a standard-definition television (SDTV) is 480 lines of resolution—but wait!

Actually, there isn't enough bandwidth in the signal to draw 480 lines per frame, so in fact, you get two alternating frames of 240 lines, which are then displayed simultaneously, *interlaced* with odd-numbered lines coming from one frame, and even-numbered lines from another. This is called "480i," with the i indicating an interlaced resolution. (Some systems don't even do 480i, doing instead only 240 lines total.)

Many DVD players, as well as some game consoles (such as the last-generation Xbox, some GameCubes, and the PS2), also support an additional option, available only when using component video cables, called "480p." The "p" stands for *progressive*, and indicates that the whole 480-line frame is drawn all at once. Some TV vendors refer to 480p support as Enhanced Definition TV (EDTV), in contrast with High Definition TV (HDTV).

Higher resolutions are 720p (720 lines, progressive scan), 1080i (1080 lines, interlaced, which is two alternating fields of 540), and 1080p (1080 lines, progressive scan). Competition among these, and budget-conscious hardware developers, mean that some devices support only some of these.

None of these resolutions inherently specify a horizontal resolution, but on the PS3, you can generally safely assume that the output will be scaled for a 16x9 display by default.

Finally, all of these resolutions include *overscan* —on a standard television, the image is normally displayed slightly beyond the edges of the display, to ensure that there are no black bars around the image, which many users find distracting. So, the default Linux handling is to use only part of the display, rather than the full theoretical resolution. For each of these video modes, there's a corresponding "full screen" video mode that actually uses all the space. Terra Soft has an excellent page on video modes (see Resources).

So, let's discuss the amended specifications. PS3 Linux runs on a virtual machine that has a Cell/B.E. processor with 6 SPEs, about 212MB of main memory, a single network interface, a SCSI disk (plus several more SCSI disks if you have a card reader), a SCSI device that may be a DVD-ROM, CD-ROM, or possibly a Blu-ray reader, and functional USB and Bluetooth ports. Graphics are provided by a special framebuffer device, which supports a broad array of video modes (see Understanding consumer electronics video modes).

Plan your development environment

To make good use of a PS3 for development, first decide what you need out of the development environment. If you are comfortable abandoning X entirely, it'll save you a huge amount of memory (the Fedora default of running Gnome is, perhaps, not one of the most memory-conserving strategies). A lot of the system setup utilities are most accessible through X, but they can take a long time to run. I ran the graphical package updater under Gnome, and it took several hours. This is probably because the process had a total memory footprint of 375MB, substantially more than the physical memory available. It spent most of its time waiting for the swap disk, thrashing like Emacs on a VAX.

The lesson here is that available memory is going to be your biggest weakness on the PS3. Turn off services you don't need as quickly as possible. Run xterm instead of gnome-terminal. If you need multiple windows, consider running *screen* instead of multiple xterms.

Display options

If you're going to run headless, don't worry too much about these. But if you want to run with a display that you're going to look at, you need to consider display options. The best quality comes from using the PS3's HDMI output. If you have a computer monitor, HDMI to DVI cables can be had quite cheaply, but be sure your monitor supports High-bandwidth Digital Content Protection (HDCP). If it doesn't, you'll get a black screen, or possibly brightly-colored static. If you're using a KVM switch, note that the switch too, not just the display, has to support HDCP! Unfortunately, there appears to be no way to turn this "feature" off; you simply can't get an unencrypted signal. While this makes some economic sense for a device that plays high-definition movies, it can be a bit frustrating to need special permission to view your X desktop. HDMI/DVI supports pretty much every video mode the PS3 has to offer, both the consumer electronics modes, such as 720p, and the PC modes, such as WXGA.

If you can't do HDCP, or don't have a DVI-capable monitor, your next best bet is component video. Component video, mostly accepted by TVs, but also by some computer monitors, is very similar to plain old VGA, only it comes out as three RCA connectors, one red, one green, and one blue. The component video cable is an additional-price addon—but it's the same as the component video cable for the PS2, if you had one of those. Component video can reliably support 720p and 1080i displays, but may not work for 1080p (see Understanding consumer electronics video modes if you're not sure what this is). Component video offers both higher resolution and better clarity than the next two options.

If you can't do component, you might have a device that supports S-video. S-video supports 480i, but divides the signal into two separate channels, luminance and chroma (that's "brightness" and "color"), giving better sharpness than your final option, composite video. Composite video is the plain old RCA jack, usually colored yellow, that is compatible with nearly everything. If you can possibly avoid composite, do.

Note that the Fedora 7 installer is essentially unusable with anything under 720p, unless you force it into text mode. Really, if you plan to use the PS3 as a development system, you ought to favor higher resolution modes. Linux systems have developed an affinity for larger displays, and the Fedora windowing environment is just plain awful on a low-resolution display. Furthermore, the lower-resolution displays, especially over composite cables, are very hard to read, with horrible blurring.

My system was set up for WXGA (1280x768) over DVI, and this is quite usable. It's enough desktop real estate for a couple of windows without overlap.

Install Fedora

By now your Fedora DVD should have downloaded, so the next step is to install it. Follow the usual PS3 instructions for getting your disk partitioned (you do lose all your data if you do this, so make backups!) and the Other OS boot loader loaded. When the time comes to put in the Fedora DVD, and boot from it, enter `linux64 text` or `linux64 video=720p` at the `kboot` prompt. 720p is the minimum usable installation video size; if you don't have component or DVI output working, use text. Booting with the default options resulted in the installer apparently freezing with a Fedora logo on screen; in fact, the installer's window was larger than the default display area, leaving me with no option but to power the machine down and try again.

After this, go ahead and get all the current updates from Fedora's update servers, and install them. This will take quite a while, as will the initial install. In both cases, once it gets started, go do something else for a couple of hours.

Next: Installing PS3 addons

Part 2 of this series discusses getting the latest PS3 addons installed and updated on your system, and some of the configuration changes you can make to reduce the basic memory footprint until you've got a bit of breathing room.

Resources

Learn

- See all parts in the *Linux development on the PlayStation 3* series.
- "Programming high-performance applications on the Cell BE processor, Part 1: An introduction to Linux on the PLAYSTATION 3" (developerWorks, January 2007) and "PS3 fab-to-lab, Part 1: Build Linux lab equipment from a Sony PLAYSTATION 3" (developerWorks, May 2007) are the first articles in other developerWorks series about running Linux on the PS3.
- Read Sony's guide to the OtherOS feature, including the download of the installer.
- Terra Soft's page listing video modes is very helpful.
- Wikipedia's Linux for PlayStation 3 is a good starting place for learning more.
- In the developerWorks Linux zone, find more resources for Linux developers, and scan our most popular articles and tutorials.

- See all Linux tips and Linux tutorials on developerWorks.
- Stay current with developerWorks technical events and Webcasts.

Get products and technologies

- The Fedora project makes it very easy to obtain DVDs.
- Get the PS3 linux distro kit (also known as the "addons CD") from this wonderful collection of interesting Cell-related things.
- At the Cell Broadband Engine Resource Center on developerWorks, you can get the Installing FC7 on PS3 (SDK 3.0) (Updated 01/15/08) infobomb.
- Order the SEK for Linux, a two-DVD set containing the latest IBM trial software for Linux from DB2®, Lotus®, Rational®, Tivoli®, and WebSphere®.
- With IBM trial software, available for download directly from developerWorks, build your next development project on Linux.

Discuss

- Get involved in the developerWorks community through blogs, forums, podcasts, and community topics in our new developerWorks spaces.

About the author

Peter Seebach has been collecting video game consoles for years, but has only been running Linux on them recently. He is still not sure whether this is a Linux machine that plays video games, or a game machine that runs Linux.

Trademarks | My developerWorks terms and conditions

Linux development on the PlayStation 3, Part 2: Working with memory

Identifying memory hogs on PS3 Linux

Peter Seebach, Freelance author, Plethora.net

Summary: The Sony PlayStation 3 (PS3) runs Linux®, but getting it to run well requires some tweaking. In this article, the second in a series, Peter Seebach takes a look at where all the memory goes and how to reclaim it.

Date: 31 Mar 2008
Level: Intermediate
PDF: A4 and Letter (34KB | 8 pages)Get Adobe® Reader®
Also available in: Russian Japanese Portuguese

Activity: 14181 views
Comments: 0 (Add comments)

★ ★ ★ ★ ★ Average rating (based on 15 votes)

In Part 1 of this series, I introduced PlayStation 3 (PS3) Linux and its strengths and weaknesses as a development environment. This second part covers some of the things that can have a significant impact on a PS3 system's performance running Linux.

Not every piece of advice here will work for everyone. If you're doing graphics, you can't just run the system without X. On the other hand, if you're not doing graphics, nothing else makes as much of a difference in the system's memory footprint.

The focus on memory may seem odd if you're used to desktop systems that have never mounted a swap drive in anger, but, in fact, the PS3 doesn't have enough memory for a modern desktop Linux to feel comfortable on it out of the box. With a default Fedora 7 install, the system spends a lot of its time (and yours) swapping. Swapping imposes a large penalty on any system; on a system built around a 2.5-inch disk drive accessed through a hypervisor, with main memory that's substantially faster than what you find in most desktops, the contrast is even sharper than it is on a more traditional desktop system.

One other note: On my test system, rebooting was unreliable under the 2.6.21 kernel that was originally installed. The 2.6.23 kernel fixed this, using either the version from the PS3 addons CD, or the version from the standard Fedora updater. In general, the kernel from the PS3 addons CD is probably your best bet.

About this series

This series of three articles looks at PS3 Linux as a prospective development environment.

This first article, Part 1, introduces the basic configuration knobs and widgets specific to the PS3, shows you how to use them effectively, and suggests the kind of trickery that might get improved performance or a more usable display.

This article and Part 3 delve into some of the performance and tuning issues that, while they might apply on any system, are particularly useful for turning your PS3 from a proof-of-concept demo into a working system.

Reducing memory usage

This topic comes and goes in importance. For most desktop Linux users, the idea that you'd need to actually do something to reduce memory usage is a distant memory. Furthermore, because processes tend to grow to fill available space, even when a machine with 64MB of main memory was considered a powerful server, there simply wasn't as much software running, and it didn't need as much memory. The PS3 is one of the systems where memory usage matters a lot, though, and Fedora 7, charming though it is, has not been designed around small-memory systems.

To reduce memory usage, start by identifying the largest consumers of memory. One easy way to do this is with `top`, which gives you a live display of processes on the system. By default, `top` shows you processes sorted by CPU consumption, which is useful for other kinds of performance tuning, but isn't the best way to track down memory hogs. Note that `top` gives you a summary of memory usage. For instance, on a PS3 with X running but several services turned off, I got this line:

Listing 1. Am I really using that much memory?

```
Mem:    219192k total,   213692k used,      5500k free,     7232k buffers
Swap:  4192956k total,        0k used,   4192956k free,    89468k cached
```

Bring up `top` (just run `top` in a shell), then hit **O** (that's a capital O, as in "order by"), then hit **q**, then hit **Return**. In this case, **q** means "resident size" and tells you how much actual memory the process is using. Another likely choice would be "virtual size" (option **o**).

You should now see a list of processes, sorted by actual physical memory usage. Here's a partial listing, again from a test machine:

Listing 2. I guess I am using that much memory

```
  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 3259 root      20   0 65424  36m 4996 S    2 17.2   0:01.39 Xorg
 3422 seebs     20   0 92900  24m  20m S    0 11.6   0:01.22 nautilus
 3439 seebs     20   0 58600  24m  22m S    0 11.5   0:00.36 nm-applet
 3473 seebs     20   0 56620  24m  14m S    0 11.4   0:01.22 /usr/bin/sealer
 3420 seebs     20   0 50248  21m  18m S    0 10.2   0:01.90 gnome-panel
 3476 seebs     20   0 48988  14m  10m S    1  6.9   0:00.64 gnome-terminal
 3445 seebs     20   0 33104  14m 9464 S    0  6.7   0:00.40 puplet
 3453 seebs     20   0 45764  13m  12m S    0  6.4   0:00.22 gnome-power-man
 3414 seebs     20   0 41920 9696 8052 S    0  4.4   0:00.29 gnome-settings-
 3418 seebs     20   0 22200 8996 7316 S    0  4.1   0:00.33 metacity
```

```
3297 seebs     20   0 40544 8384 7088 S    0  3.8   0:00.32 gnome-session
3432 seebs     20   0 20076 6120 5244 S    0  2.8   0:00.10 bluetooth-apple
3444 seebs     20   0 14692 6060 3532 S    0  2.8   0:00.24 python
```

The top ten or so consumers of memory are all X-related. This is why, if you really want to free up memory, one of your first choices might be to shut down X. Noticing how many of those applications are GNOME-specific, you might be tempted to try KDE, but I'm afraid that won't get you anywhere. KDE has a comparable memory footprint on the PS3.

In fact, if you absolutely need X, your best option is not to use the X session environments offered by the X login window; instead, log in on the console and start X with a smaller window manager and fewer additional programs. But how to do that? Your first step will be to exit `top` and get your prompt back—just hit **q**.

Making use of runlevels

Runlevels are a feature that many Linux users never have cause to learn about. They're essentially inherited from System V UNIX®, although there are some differences, of course. A *runlevel* is a defined set of system services that are run together. For historical reasons, the usual desktop Linux environment, with a graphical login program that spawns Gnome or KDE, is called runlevel 5. A conventional standalone workstation without X would often run in runlevel 2. In theory, you can do pretty well by just changing the system's runlevel directly with the `init` command, found in /sbin. For instance, running (as root) `/sbin/init 2` tells the system to move to runlevel 2. (Usually this is done by stopping any services not used in runlevel 2, then starting any services that are used in runlevel 2.)

The default runlevel is set by a line in the /etc/inittab file, which looks like this:

Listing 3. A default runlevel

```
id:5:initdefault:
```

Runlevels

There are actually 7 runlevels, 0 through 6. 0 is "halt," and is the same behavior you get from telling the system to shut down. 1 is "single user mode," which is used mostly for diagnostics and repairs. 2 is multiuser, and 3 is multiuser "plus networking," which mostly means NFS. 4 is unused, 5 is XDM, and 6 is reboot.

As the /etc/inittab file documents, you do not want to ever set the default runlevel of a system to 0 or 6. It can be hard to recover from this, although a friendly boot loader or a recovery CD will let you get in and fix it. The default for nearly all modern desktop Linux systems that use runlevels is 5. Some systems use other startup mechanisms, but runlevel 5 is the one found in Fedora 7.

The format of the line is historical, and all you really need to know to change it effectively is that you can change the 5 to a 2 or a 3. Then the next time the system boots, it will come up to a text console login prompt rather than starting X.

The simple fact is, the text console is a much better choice for a development system with limited memory than the full-bore X environment. This is a good time to do some poking around to see what we can get rid of. Change the `initdefault` value in your /etc/inittab to 3 and reboot. (You can, in fact, use the `init` command to transition, but there are edge cases where this doesn't work.) You'll immediately notice how much more quickly the system comes up. Immediately after logging in, run `top` again. Results will vary, but you might see that memory is a perhaps half full, instead of nearly completely full. Progress!

That's still a lot of memory

While it's certainly good to have a hundred megabytes free, the system could perhaps be slimmed down a bit more. Another run through `top` reveals some more memory hogs:

Listing 4. Excuse me, I don't think I ordered that

```
  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 2204 root      39  19 30056  12m 5632 S    0  5.7   0:00.70 yum-updatesd
 1825 root      20   0 19220 5052  972 S    0  2.3   0:00.00 python
 2238 haldaemo  20   0  6648 2728 2180 S    0  1.2   0:00.27 hald
 1909 root      20   0 13480 2236 1564 S    0  1.0   0:00.02 cupsd
 2015 root      20   0 12352 1976 1272 S    0  0.9   0:00.02 console-kit-dae
 1958 root      20   0 12132 1796  748 S    0  0.8   0:00.01 sendmail
 2301 root      20   0  5548 1792 1484 S    0  0.8   0:00.02 login
 2141 xfs       20   0  5388 1776  884 S    0  0.8   0:00.05 xfs
 2425 seebs     20   0  5392 1732 1480 S    0  0.8   0:00.08 bash
 2371 seebs     20   0  5392 1728 1480 S    0  0.8   0:00.08 bash
 1966 smmsp     20   0 10356 1576  696 S    0  0.7   0:00.00 sendmail
 2221 avahi     20   0  3772 1520 1344 S    0  0.7   0:00.07 avahi-daemon
 1751 root      20   0 11332 1348  588 S    0  0.6   0:00.43 pcscd
 1796 root      20   0 29652 1304 1032 S    0  0.6   0:00.02 automount
```

By far, the largest program is yum-updatesd, presumably a daemon related to the system's yum updater. (In the dot com era, that kind of deep insight could have gotten you relocated anywhere in the country you wanted to go.) Luckily, this is also a program we can easily do without; you can run yum by hand when you feel like it.

Editing runlevels

Sadly, there's not a specific runlevel for "runlevel 3, only without yum-updatesd". That means it's time to start manually removing services. There are a couple of ways to do this. Each runlevel is defined by a corresponding directory in /etc/rc.d, named rcN.d; for instance, runlevel 3 is defined by the files in /etc/rc.d/rc3.d. (There are also symlinks to these directories in /etc, on a Fedora 7 system, but it's a good habit to use the full path.)

Each such directory contains a number of files, with slightly cryptic names like "K74nscd" or "S88nasd." The naming convention is simple: names starting with a K are services to be stopped when entering this runlevel (presumably from a higher-numbered runlevel, which might have been using them), and names starting with an S are services to be started when entering this runlevel. The two-digit numbers are used to sort services; S13rpcbind is started before S14nfslock, which is in turn started before S25netfs. Simple, and effective.

In fact, these are usually not files, but symlinks to scripts stored in /etc/rc.d/init.d. Typically, there's a single script that can either start or stop a given service, and then links to it are made appropriately. When `init` changes levels, it calls the scripts with "start" or "stop" arguments, as indicated.

If you're feeling like just diving in, guns blazing, you can simply remove unwanted S or K links from a runlevel directory. Similarly, you can add new links. Another option is to use the `chkconfig` utility; this is a very flexible and powerful utility that can maintain these symlinks for you. A caveat: if you remove something crucial, you may have to go in with a rescue CD of some sort (such as the Fedora install disc; see Resources) to get your system booting cleanly again. Be sure you understand what things are, and what depends on them, before you remove them!

As an example, if you wanted to remove the yum-updatesd program from runlevel 2, you could simply remove the link /etc/rc.d/rc2.d/S97yum-updatesd. To remove it from runlevel 2 with `chkconfig`, the command would be `/sbin/chkconfig yum-updatesd off`.

Tracking down more space

With `chkconfig` and `top`, it's possible to track down a fair number of large space users, figure out what they provide, and remove them if you don't need them. But what about the Python process? There's no Python service. The `ps` command reveals more:

Listing 5. Spying in Python

```
$ ps ax | grep python
 1825 ?        S      0:00 python ./hpssd.py
```

A quick `grep` among the startup scripts reveals that this is part of the hplip service, which provides "HP Linux Imaging and Printing." This can be used with some HP printers and scanners, but is otherwise unneeded. So, if you haven't turned that off already, do it now.

On a fairly typical dedicated development system, the final total was 49,896KB used, down from an initial start of 213,692KB. Free memory went from 5,500KB to 169,296KB—a noticeable improvement in room to run the compiler in. The difference this makes will vary depending on your workload; many of the background daemons, once swapped out, will stay swapped out and leave your system nearly as responsive as it would be without them. Over a long period of time, though, even a smallish difference in compile times adds up.

Next: Getting a usable X environment

As you see, if you're willing to sacrifice a number of unnecessary or unused features, you can reclaim a huge amount of system memory, leaving you with plenty of memory to run compilers and start developing code. However, many users will find the complete loss of X too high a price to pay. Part 3 in this series looks at what you can do to get a usable X environment for doing simple graphical work, without losing the ability to run the compiler.

Resources

Learn

- See all parts in the *Linux development on the PlayStation 3* series.
- "Programming high-performance applications on the Cell BE processor, Part 1: An introduction to Linux on the PLAYSTATION 3" (developerWorks, January 2007) and "PS3 fab-to-lab, Part 1: Build Linux lab equipment from a Sony PLAYSTATION 3" (developerWorks, May 2007) are the first articles in other developerWorks series about running Linux on the PS3.
- The Cell Broadband Engine Resource Center is the definitive resource for all things Cell/B.E.
- Learn more about top, ps, and other fine UNIX utilities from Wikipedia. For `chkconfig`, see man chkconfig.
- Wikipedia also has the skinny on swap.
- Your Fedora install CD has a rescue mode which may be able to help you out of a jam.
- In the developerWorks Linux zone, find more resources for Linux developers, and scan our most popular articles and tutorials.
- See all Linux tips and Linux tutorials on developerWorks.
- Stay current with developerWorks technical events and Webcasts.

Get products and technologies

- Order the SEK for Linux, a two-DVD set containing the latest IBM trial software for Linux from DB2®, Lotus®, Rational®, Tivoli®, and WebSphere®.
- With IBM trial software, available for download directly from developerWorks, build your next development project on Linux.

Discuss

- Get involved in the developerWorks community through blogs, forums, podcasts, and community topics in our new developerWorks spaces.

About the author

Peter Seebach has been collecting video game consoles for years, but has only been running Linux on them recently. He is still not sure whether this is a Linux machine that plays video games, or a game machine that runs Linux.

Trademarks | My developerWorks terms and conditions