２００５情財第１１６９号

２００５年度　オープンソースソフトウェア
活用基盤整備事業

ＯＳＳを用いた計算機システムの信頼性向上を目指した
イベントトレース機能の開発

システムコールトレーサ
使用説明書

２００６年８月

（株）日立製作所

— 目 次 —

1. はじめに

1.1. 開発の目的

近年、エンタープライズ向けシステムへの Linux の適用が増えてきている。エンタープライズ向けでは、障害でのダウンタイムを極力少なくすることが求められており、万が一障害が発生した際には迅速に原因を解明することが必要となる。

Linux では、障害解析のために、アプリケーションが発行するシステムコールの情報の引数や戻り値を収集できるツール、strace が提供されている。システムコールはシステムの挙動を把握するのに有用な情報であり、障害解析にも役に立つため、starce は広く利用されている。strace はユーザアプリケーションであり、カーネルのソースコードを変更することなく手軽にシステムコール情報を入手することができる。しかしその反面、システムが高負荷になったりクラッシュしたりすると情報を得ることが出来ない、というデメリットがある。すなわち、現在の strace ではこの高負荷やクラッシュした原因を突き止めるための情報を提供できない。

そこで、以下を目的として、カーネル空間でシステムコール情報を収集できる機能、システムコールトレーサの開発を行った。

（1）カーネルの変更が不要なこと

カーネル空間でシステムコールを記録する機能として、現在 LKST（＊）などが公開されている。しかし、LKST を利用するにはパッチを当ててカーネルを変更しなければならない。今回はシステムコールに限定し、情報を記録する機能をカーネルを変更せずに実現する。

＊）2004 年度下期、および 2005 年度上期のオープンソースソフトウェア開発基盤整備事業の一環で開発を実施。

（2）取得した情報をカーネル空間に保存すること

システムがパニックした場合にも、メモリダンプから実行していたシステムコール収集し、障害解析できるようにするため、実行システムコールの情報をカーネル空間のバッファに保存する方式を採る。

（3）ユーザ空間の負荷に左右されずに情報を収集すること

システムが過負荷になりアプリケーションが正常に動作しない場合でも情報を採取することができるように、カーネル空間で情報収集を行う機能を開発する。

1.2. 稼動要件

今回開発するシステムコールトレーサの稼動要件は、以下の通りとする。

（1）ハードウェア要件：

（a）IA-32 マシン

（b）IA-64 マシン

（2）ソフトウェア要件：

（a）ディストリビューション

- Red Hat Enterprise Linux AS 4 Update-1(RHEL4-U1)
  （カーネルバージョン：kernel-2.6.9-11.ELsmp）

（b）必要なソフトウェアパッケージ

- RHEL４のカーネルパッケージ（kernel-devel-2.6.9-11.EL）

2. システムコールトレーサ（kstrax）概要

システムコールトレーサ（以下 kstrax と称する）は、カーネルのソースコードを変更することなくカーネル空間でシステムコールを記録し、ユーザ空間からその情報を取得するツールである。

2.1. 提供機能

kstrax v1.0.0 は以下の機能を提供する。

（1）以下のシステムコール情報をカーネル空間内で記録する機能
- （ｉ）システムコールの入り口
  - （ａ）システムコール番号
  - （ｂ）システムコールの開始時間
  - （ｃ）システムコールの引数
  - （ｄ）システムコールを発行したプロセスのプロセス識別子
  - （ｅ）open/create したファイルの名前
- （ii）システムコールの出口
  - （ａ）システムコール番号
  - （ｄ）システムコールを発行したプロセスのプロセス識別子
  - （ｆ）システムコールの終了時間
  - （ｇ）システムコールの返り値

（2）カーネル空間内で記録したデータをユーザ空間へコピー（バイナリファイル出力）する機能

（3）（1）（2）の制御、および取得したシステムコール情報を解析して表示する機能
- （ａ）時系列表示（RAW モード）：テキスト
- （ｂ）実行開始・終了対応型表示：テキスト、グラフ
- （ｃ）統計表示：テキスト、グラフ

2.2. ソフトウェア構成

kstrax v1.0.0 は、次の 2 つのソフトウェアから成る（図 2-1 参照）。

（1） kstrax モジュール

このモジュールはカーネルモジュールであり、以下のモジュールからなる。

(a) ctr_mod：システムコールテーブルを書き換える。現在、このモジュールは一度ロードしたら、アンロード不可である。アンロードを行うとカーネルパニックを起こす可能性がある。

(b) kstrax_buf：システム内で発行されたシステムコールの情報をカーネル空間で記録する。このモジュールはアンロード可能であり、アンロードするとカーネル空間での記録を停止する。

(c) kstrax_stub（IA-64 版のみ）:スタブを構成する。IA-32 版では同様の機能は ctr_mod に含まれている。本モジュールもアンロード不可である。

（2） kstrax コマンド群

このコマンド群は次の機能を提供する。

(a) カーネル空間に記録されたシステムコール情報を、ユーザ空間内にコピー（バイナリファイル出力）するための操作コマンド

(b) バイナリデータを整形し、テキストファイルへ出力するコマンド

(c) テキストファイルをフィルタリングするコマンド

(d) バイナリデータから、グラフ表示を行なうコマンド

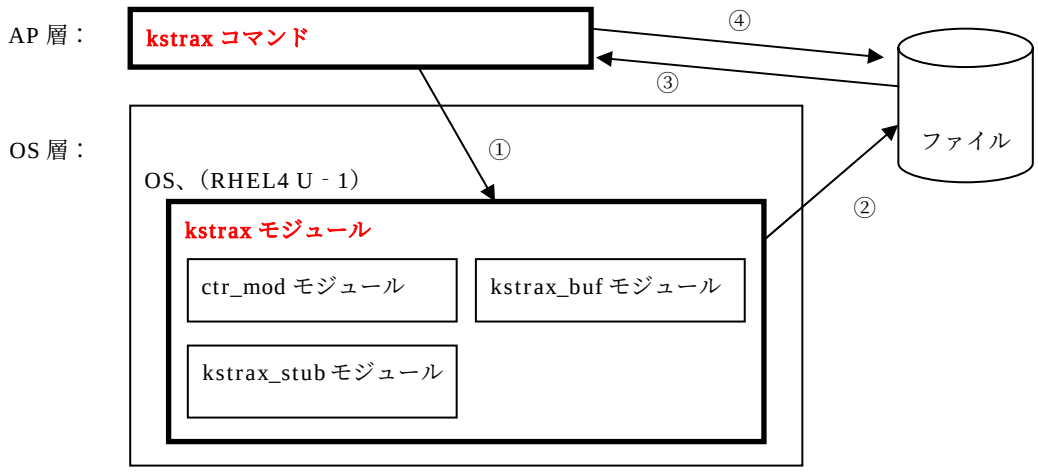


図 2-1 ソフトウェア構成図

3. kstrax インストール方法

kstrax v1.0.0 をインストールする手順を以下に示す。尚、例として IA-32 の場合を示す。基本的に IA-64 でも同様である。

（1）kstrax のアーカイブ（ファイル名：kstrax-1.0.0.tar.bz2）を取得し[1]、任意のディレクトリ（ここでは**$(SOMEWHERE_LIST)**）に解凍する。

```
$ cd $(SOMEWHERE_LIST)
$ tar –xjf kstrax-1.0.0.tar.bz2
```

（2）モジュールとユーザ空間コマンドを作成する。

（a）スーパーユーザでログインし直す。

```
$ su
Password:　（スーパーユーザのパスワードが要求されるので、ここで入力する）
```

（b）（1）で展開した kstrax ディレクトリへ移動し、ビルドする。

```
# cd $(SOMEWHERE_LIST) / kstrax /
# make
```

（c）module ディレクトリに kstrax_buf.ko、module/arch/i386(ia64)ディレクトリに ctr_mod.ko（kstrax_stub.ko：IA64 のみ）が作成されていることを確認する。（図 3-1 参照）

```
# ls module
# ls module / arch / i386
```

（d）src ディレクトリ以下に kstrax が作成されていることを確認する。（図 3-2 参照）

```
$ ls src
```

[1] sourceforge からダウンロード可（http://sourceforge.jp/projects/kstrax/）

（３）モジュールとユーザ空間コマンドをインストールする。

（ａ）スーパーユーザでログインし直す。

$ su

Password:　（スーパーユーザのパスワードが要求されるので、ここで入力する）

（ｂ）（１）で展開したディレクトリへ移動してインストールを行なう。

 # cd $(SOMEWHERE_LIST) / kstrax /

 # make install

（ｃ）/lib/modules/$(SOME VERSION)/kernel/drivers/kstrax 以下にモジュールがインストールされていることを確認する。（図 3-3 参照）

$ ls /lib/modules/$(SOME VERSION)/kernel/drivers/kstrax

（ｄ）/usr/sbin 以下にユーザ空間コマンドがインストールされていることを確認する。（図 3-3 参照）

$ ls /usr/sbin/kstrax*

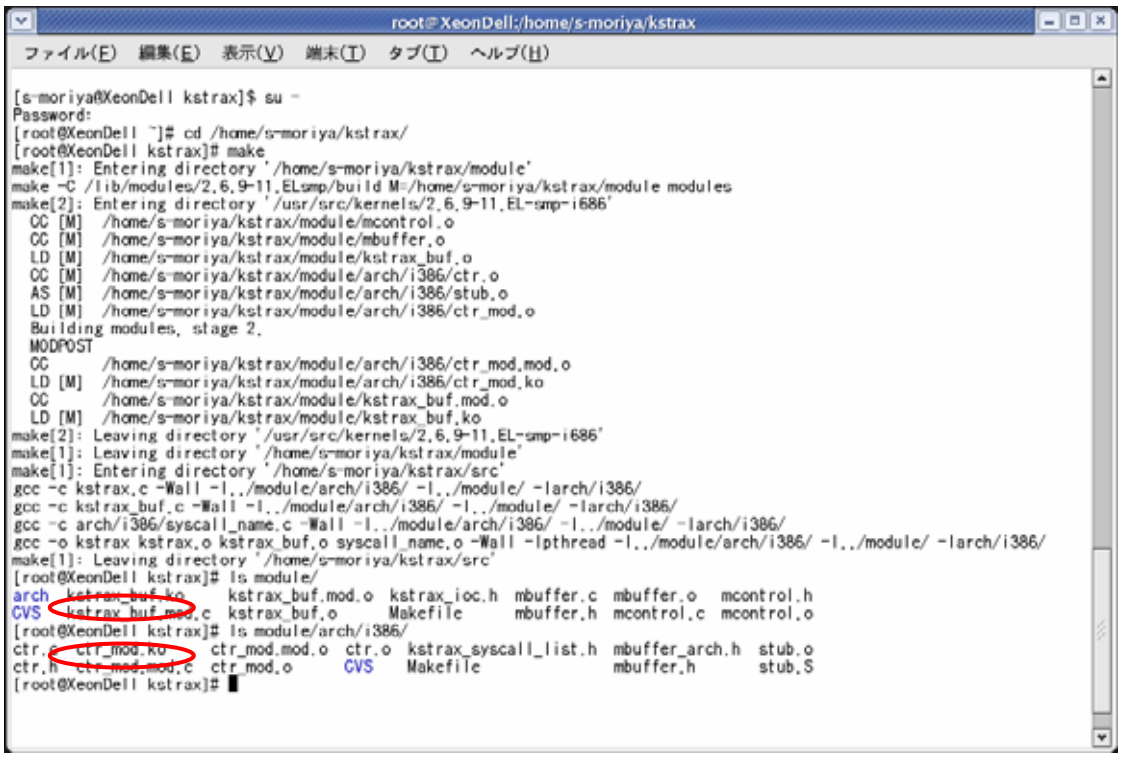


図 3-1 kstrax モジュール生成

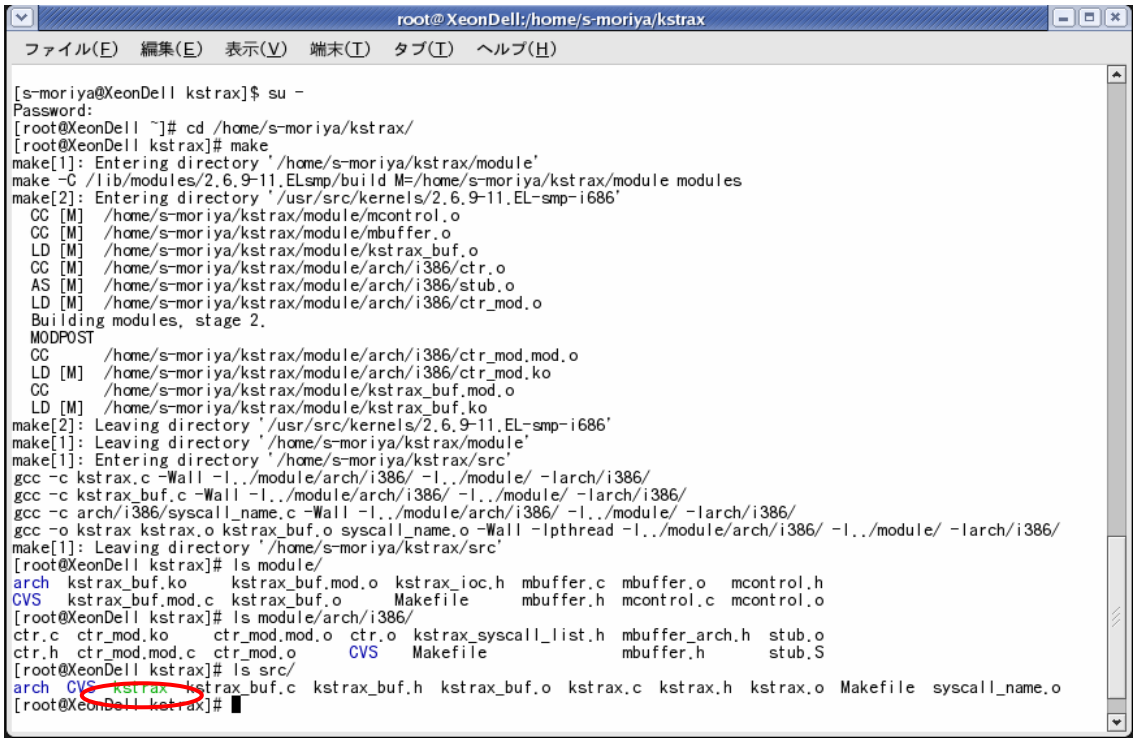


図 3-2 kstrax コマンド生成

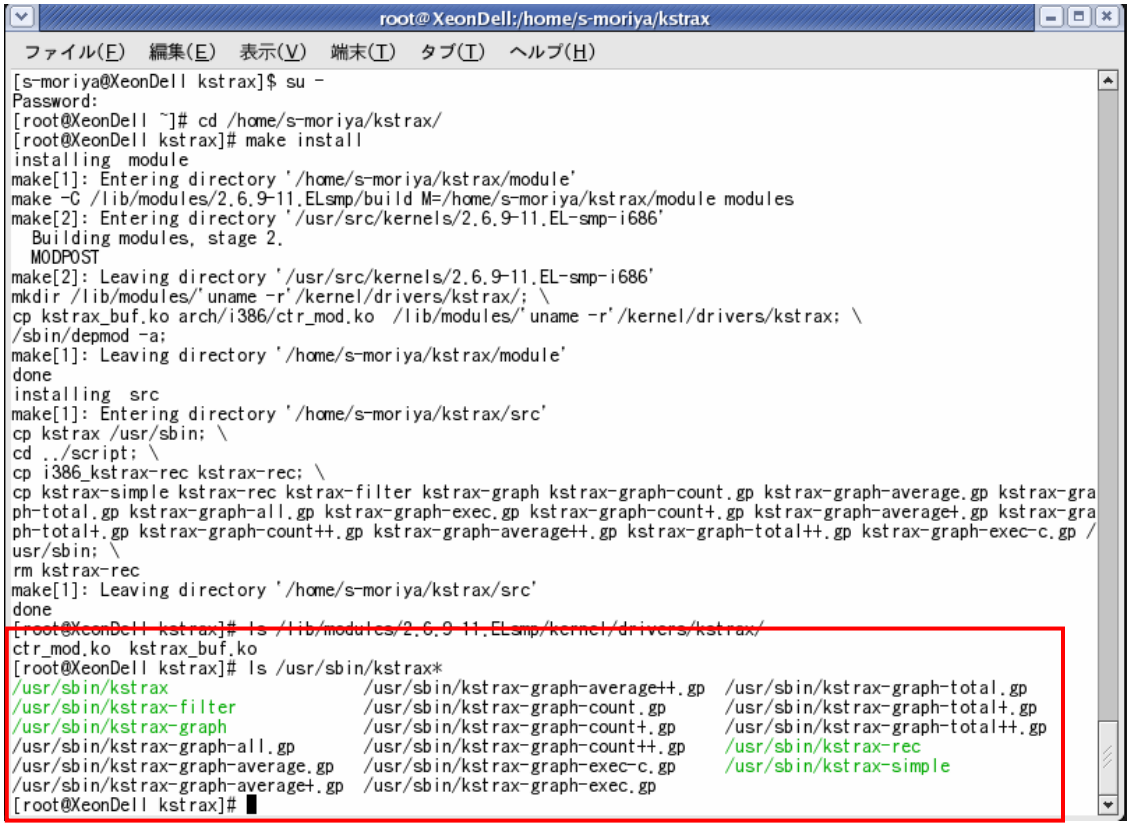


図 3-3 kstrax インストール後

4. 使用方法

4.1. 簡易スクリプトなし

（1）カーネル空間でシステムコールを記録（モジュールのロード）

（a）スーパユーザでログインし直す。

$ su

Password: （スーパーユーザのパスワードが要求されるので、ここで入力する）

（b）モジュールをロードする（詳細は 5 **マニュアル**参照)。

 # kstrax-rec start <parameter>

（c）ctr_mod モジュール、kstrax_buf モジュール、（kstrax_stub モジュール：ia64 のみ）がロードされたことを以下のコマンドで確認する。（図 4-1 参照）

 # lsmod

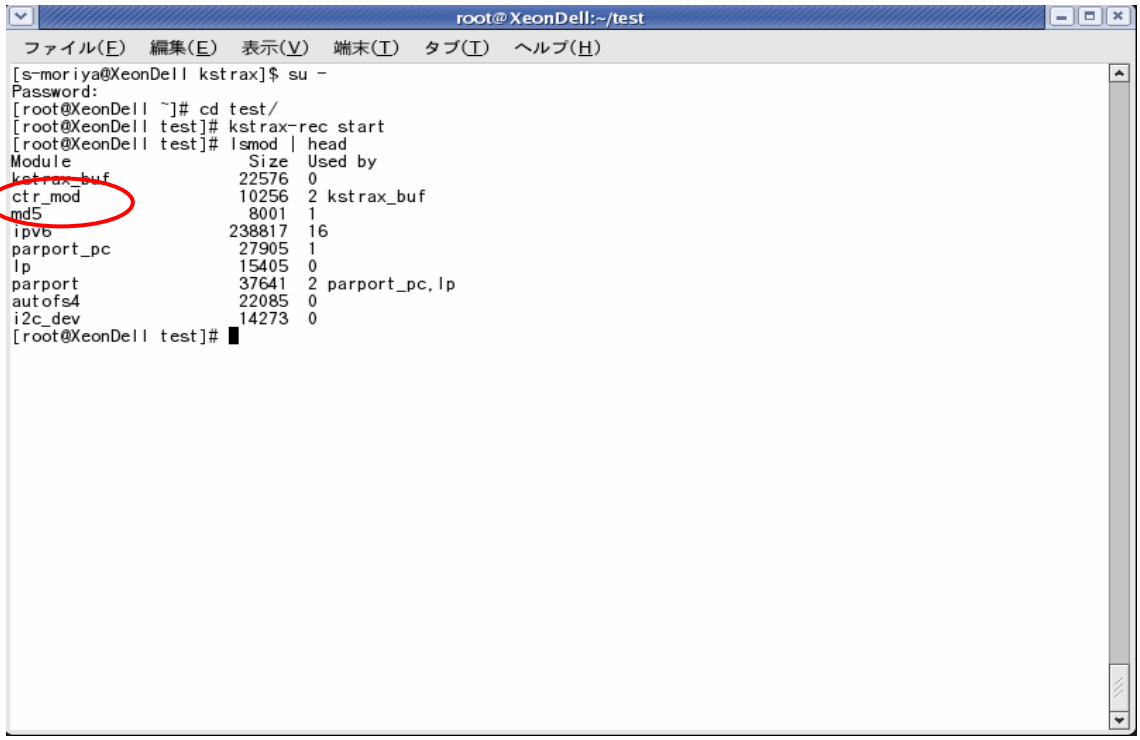


図 4-1 ロードの確認

（2）システムコール情報をバイナリファイルへ出力

（a）スーパユーザでログインし直す。

$ su

Password: （スーパーユーザのパスワードが要求されるので、ここで入力する）

（b）下記コマンドを入力する。尚、本コマンドの終了は Ctrl + C で行う。（詳細は５マニュアル参照）

$ kstrax –b <option>

Ctrl + C →終了

（3）バイナリファイルデータを整形しテキストファイルへ出力

（2）で作成したバイナリファイルがあるディレクトリへ移動し、下記コマンドを入力する。なお、出力ファイルの最後では出力されるシステムコールが欠落する可能性がある。（詳細は５５．マニュアル参照）

$ cd $(SOMEWHERE_LIST)

$ kstrax –t <filename> <option>

（4）カーネル空間でのシステムコール記録を終了（モジュールのアンロード）

（a）スーパユーザでログインし直す。

$ su

Password: （スーパーユーザのパスワードが要求されるので、ここで入力する）

（b）モジュールをアンロードする。

 # kstrax-rec stop

（c）kstrax_buf モジュールがアンロードされたことを以下のコマンドで確認する。（図 4-2 参照）

 # lsmod

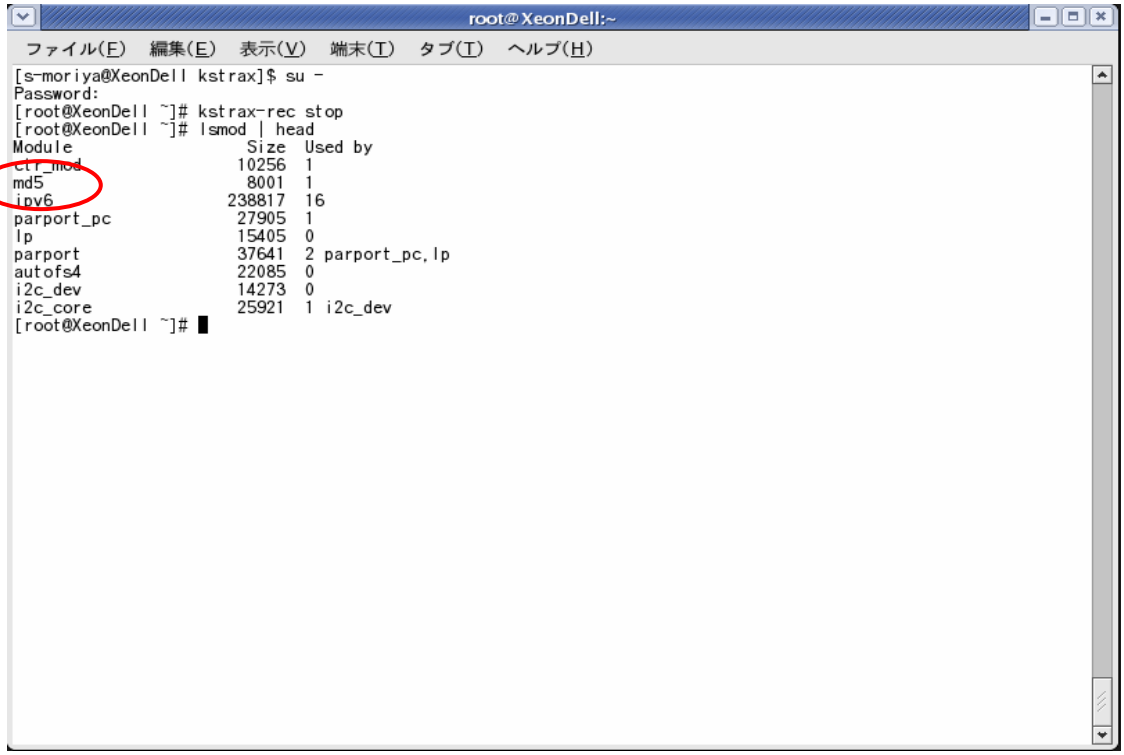


図 4-2　kstrax_buf モジュールのアンロード確認

4.2.　簡易スクリプト利用時

簡易スクリプトによりモジュールのロード＋バイナリデータ出力、バイナリデータ出力停止＋モジュールアンロードをそれぞれひとつのコマンドで実行できる。

（1）スーパユーザでログインし直す。

$ su

Password:　（スーパーユーザのパスワードが要求されるので、ここで入力する）

（2）トレースデータ出力

簡易操作用スクリプトを実行（詳細は５．マニュアルを参照)。

$ kstrax-simple start　<parameter>

（3）トレース終了

簡易操作用スクリプトを実行（詳細は５．マニュアルを参照)。

$ kstrax-simple stop

（4）バイナリデータからテキストデータへの変換は 4.1 と同様

5. マニュアル

5.1. kstrax-rec コマンド（モジュールロード／アンロードコマンド）

5.1.1. コマンドライン文法

kstrax-rec [option] <parameter>

5.1.2. 概要

kstrax-rec コマンドは、以下の 2 つの機能を提供する

（1） カーネル空間での記録の開始（＝ctr_mod モジュール、kstrax_buf モジュール、（kstrax_stub：ia64 のみ）モジュールのロード）
これらのモジュールがロードされた時点から、システムコールの記録が開始される。
本コマンドはカーネル内の記録バッファの初期化も行う。

（2） 記録の終了（＝kstrax_buf モジュールのアンロード）
kstrax_bufモジュールがアンロードされた時点でシステムコールの記録は中止となる。

一度終了した記録を再開する場合には、（1）の制御を実施するが、記録用のカーネルバッファは初期化されるので、以前のデータはすべて消えていることに注意。

5.1.3. 詳細仕様

（1） option

(a) start：カーネル空間でのシステムコールの記録を開始する。（表 5-1 参照）

(b) stop：カーネル空間でのシステムコールの記録を終了する。（表 5-2 参照）

（2） parameter

システムコールを記録するカーネル空間バッファの大きさ（エントリ数）を設定する。オプションで start を選択した場合のみ有効。

指定がない場合は、デフォルト値として、各 CPU において 160000 エントリ（6.4MB—IA32 / 12MB—IA64）が確保される。

指定できる範囲は以下の通り。

- IA32 の場合
  - ✧ 最小値：各 CPU において 1 エントリ（40B）
  - ✧ 最大値：各 CPU において 2500000 エントリ/CPU 数
- IA64 の場合(vmalloc 領域はページサイズ 4KB のとき 1TB ある)
  - ✧ 最小値：各 CPU において 1 エントリ（74B）
  - ✧ 最大値：各 CPU において 14G エントリ/CPU 数（ページサイズ 4KB）
  - ✧ 最大値：各 CPU において 14G*16 エントリ/CPU 数（ページサイズ 8KB）
  - ✧ 最大値：各 CPU において 14G*16*16 エントリ/CPU 数（ページサイズ 16KB）
  - ✧ 最大値：各 CPU において 14G*16*16*16*16 エントリ/CPU 数（ページサイズ 64KB）

表 5-1 記録開始

項目	説明
オプション名	記録開始
文法	start <parameter>
説明	カーネル空間でのシステムコールの記録を開始する。parameter はカーネル空間バッファの大きさを設定する（デフォルト値は 160000 エントリ）

表 5-2 記録終了

項目	説明
オプション名	記録終了
文法	stop
説明	カーネル空間でのシステムコールの記録を終了する。

5.2. kstrax コマンド（システムコール情報表示コマンド）

5.2.1. コマンドライン文法

kstrax [option(s)]

5.2.2. 概要

kstrax コマンドは、以下の機能を提供する。

（1） カーネル空間のバッファデータをユーザ空間にコピー（バイナリファイル出力）

（2） バイナリデータを整形しファイルへ出力

（3） Usage の表示

5.2.3. 詳細仕様

option は本コマンドの動作を指定する。

- －b：バイナリファイル出力
- －s：ユーザ空間バッファサイズ設定。－b オプション指定時にのみ有効
- －k：カーネル空間バッファクリア禁止。－b オプション指定時にのみ有効
- －T：指定した時間だけユーザ空間に出力する。－b オプション指定時にのみ有効
- －R：カーネルバッファのスナップショットを出力。－ｂオプション指定時のみ有効
- －e：システムコールの種類を指定して記録
- －p：プロセスを指定してシステムコールを記録
- －t：指定したバイナリファイルの内容から、システムコールの入口と出口のペアを探して出力。
- －r：指定したバイナリファイルの内容から RAW モード出力。システムコールの入り口と出口のペアを探さず、トレースデータをそのまま出力。
- －c：指定したバイナリファイルの内容から、システムコールの統計情報を出力。
- －o：出力ファイル名を指定
- －h：Usage の表示
- －v：バージョン情報
- －S：トレースの状態を表示（カーネルバッファサイズ／システムコール指定／プロセス指定）

詳細は表 5-3～表 5-16 を参照。

表 5-3 バイナリファイル出力

項目	説明
オプション名	バイナリファイル出力
文法	－b
説明	カーネル空間内のバッファデータをバイナリ形式でファイルへ出力する。デフォルトのファイル名は日時_ホスト名_cpu 番号である。

表 5-4 ユーザ空間バッファサイズ設定

項目	説明
オプション名	ユーザ空間バッファサイズ設定
文法	－s　< size >
説明	ユーザ空間に読み出す時に使用するユーザ空間バッファの大きさを size で指定したエントリ数で確保する。－b オプションを指定時にのみ有効。

表 5-5 カーネル空間バッファクリア禁止

項目	説明
オプション名	カーネル空間バッファクリア禁止
文法	－k
説明	バイナリファイル出力時にカーネル空間バッファをクリアしない。－b オプションを指定時にのみ有効。

表 5-6 コピー時間指定

項目	説明
オプション名	コピー時間指定
文法	－T　<sec>
説明	カーネル空間からユーザ空間へのバッファ情報のコピーを sec で指定した時間（秒）だけ行なう。－b オプションを指定時にのみ有効。

表 5-7 スナップショット出力

項目	説明
オプション名	スナップショット出力
文法	－R
説明	現在カーネルバッファに記録されている内容をバイナリファイルへ出力して終了する。

表 5-8 システムコール指定

項目	説明
オプション名	システムコール指定
文法	－e　< type >
説明	type で指定した種類のシステムコールをカーネル空間で記録する。type としては FILE、NETWORK、IPC、PROCESS、SIGNAL、ALL の他、個々のシステムコール名でも指定できる。FILE はファイル関連のシステムコール、NETWORK はネットワーク関連のシステムコール、IPC は IPC 関連のシステムコール、PROCESS はプロセス管理に関連するシステムコール、SIGNAL はシグナル関連のシステムコールをそれぞれ記録する。指定したシステムコールの記録をオフにする場合は、再度本オプションで同じシステムコールを指定する。

表 5-9 プロセス指定

項目	説明
オプション名	プロセス指定
文法	－p　< pid >
説明	pid で指定したプロセスが発行したシステムコールのみカーネル空間で記録する。pid に－1 を指定すると全プロセスをトレースする。

表 5-10 テキスト出力

項目	説明
オプション名	テキスト出力
文法	－t < filename >
説明	－b オプションで出力した filename という名前のバイナリファイルのデータからシステムコールの入口と出口の記録のペアを作成し整形して、ファイルへ出力する。デフォルトのファイル名は filename_txt となる。filename には実際に存在するファイル名（filename_cpu 番号）から_cpu 番号を取り除いた部分を入力する。出力形式を以下に示す。 **pid 開始時間 終了時間 システムコール（名前、引数、返り値）** 時間表示フォーマットは<hh：mm：ss．uu>である。また、引数、返り値は 16 進数で表示する。引数がポインタだった場合はポインタの値を出力するが、open と creat のみポインタからファイル名を求め表示する。返り値がエラーの場合は－1 とエラー内容を表示する。尚、システムコールを記録できなかった場合、記録できなかったシステムコール数を出力する。詳細は 6 **付録：kstrax の使用例**を参照。

表 5-11 RAW モード出力

項目	説明
オプション名	RAW モード出力
文法	－r < filename >
説明	－b オプションで指定した filename という名前のバイナリファイルデータからシステムコールの入口と出口のペアを作成しないでテキスト出力する。デフォルトのファイル名は filename_raw となる。出力形式を以下示す。 **serial cpuid pid 時間 システムコール（番号、名前、引数／返り値）** 時間表示フォーマットは<hh：mm：ss．uu>である。また、引数、返り値は１６進数で表示する。引数がポインタだった場合はポインタの値を出力するが、open と creat のみポインタからファイル名を求め表示する。返り値がエラーの場合は－1 とエラー内容を表示する。尚、システムコールを記録できなかった場合、記録できなかったシステムコール数を出力する。詳細は 6 **付録：kstrax の使用例**を参照。

表 5-12 統計情報出力

項目	説明
オプション名	統計情報出力
文法	－c　< filename >
説明	－b オプションで指定した filename という名前のバイナリファイルデータから記録したシステムコールの統計情報を出力する。出力先は filename_stat となる。出力形式を以下示す。 **システムコール番号、名前、呼出回数、処理時間（平均、最大、最小、総和)、エラー回数、呼出エントリがない回数、戻りエントリがない回数** 最後に、記録できなかったシステムコール数を出力する。詳細は 6 **付録：kstrax の使用例**を参照。

表 5-13 出力ファイル指定

項目	説明
オプション名	出力ファイル指定
文法	－o　<filename>
説明	－b、－t、－r、－c オプションで出力するときに、filename で指定したファイルへ出力する。詳細は 6 **付録：kstrax の使用例**を参照。

表 5-14 Usage

項目	説明
オプション名	Usage
文法	－h
説明	Usage の表示

表 5-15 バージョン情報

項目	説明
オプション名	バージョン情報
文法	－v
説明	kstrax コマンドのバージョン情報を表示

表 5-16 トレース状態情報

項目	説明
オプション名	トレース状態情報
文法	－S
説明	kstrax のトレース状態情報を表示。具体的には、カーネル空間のバッファの大きさ、トレース中のシステムコールの種類およびプロセスを表示する。出力形式を以下に示す。 kernel buf entry：------- tracing pid　　：xxx traceing syscall：yyy 詳細は 6 **付録：kstrax の使用例**を参照。

5.3. フィルタリング用スクリプト

5.3.1. コマンドライン文法

kstrax-filter [option] <parameter>

5.3.2. 概要

kstrax-filter コマンドは、以下の機能を提供する。

（1） PID によるフィルタリング
（2） システムコール名によるフィルタリング
（3） 返り値によるフィルタリング（通常終了 or エラー終了）
（4） 時間（xx より前、xx より後、xx 台）
（5） ファイル名かどうかによるフィルタリング（開いたファイル名のみ）

尚、フィルタリング結果はコンソールへ出力する。ファイルへ出力したい場合はリダイレクト機能を利用する。

5.3.3. **詳細仕様**

（1） option

- －p：pid によるフィルタリング
- －e：システムコール名によるフィルタリング
- －r：返り値によるフィルタリング
- －t：時間によるフィルタリング
- －f：ファイル名かどうかでフィルタリング

（2） parameter

フィルタリングを適用するファイルを指定する。

詳細は表 5-17～表 5-20 を参照

表 5-17 フィルタリング（キー：PID）

項目	説明
オプション名	フィルタリング（キー：PID）
文法	－p <pid>
説明	pid で指定したプロセス ID を持つシステムコールの情報のみコンソールへ表示する。

表 5-18 フィルタリング（キー：システムコール名）

項目	説明
オプション名	フィルタリング（キー：システムコール名）
文法	－e <syscall name>
説明	syscall_name で指定したシステムコール情報のみコンソールへ表示する。

表 5-19 フィルタリング（キー：返り値）

項目	説明
オプション名	フィルタリング（キー：返り値）
文法	－r <error>
説明	－r のみの場合は通常終了したシステムコールを表示する。－r のあとに error を与えた場合はエラー終了したシステムコールのみ表示する。

表 5-20 フィルタリング（キー：時間）

項目	説明
オプション名	フィルタリング（キー：時間）
文法	－t <time> <－A|－B>
説明	time で指定した時間に発行されたシステムコールの情報だけ表示する。－A を追加することで指定した時間以降のシステムコール情報を表示する。一方、－B を追加すると指定した時間以前のシステムコール情報を表示する。

表 5-21 フィルタリング（キー：ファイル名/システムコール）

項目	説明
オプション名	フィルタリング（キー：ファイル名/システムコール）
文法	－f
説明	開いたファイル名のみ表示する。

5.4. グラフ表示用スクリプト

5.4.1. コマンドライン文法

kstrax-graph [option] <parameter>

5.4.2. 概要

kstrax-graph コマンドはシステムコール情報のグラフ表示を行なうスクリプトである。

5.4.3. 詳細仕様

（1） option

- ALL：システムコールの統計情報全て（呼出回数、平均実行時間、トータル実行時間）をグラフ表示する。
- CNT：呼出回数のグラフを表示する。
- CNT+：呼出回数と平均実行時間のグラフを表示する。
- CNT++：統計情報全て含むグラフを表示する。
- AVE：平均実行時間のグラフを表示する。
- AVE+：呼出回数と平均実行時間のグラフを表示する。
- AVE++：統計情報全てを含むグラフを表示する。
- TOTAL：トータル実行時間のグラフを表示する。
- TOTAL+：呼出回数とトータル実行時間のグラフを表示する。ト
- TOTAL++：統計情報全てを含むグラフを表示する。
- EXEC：発行されたシステムコールを時系列で表示する。
- EXEC-C：発行回数が多いシステムコール 20 個について、時系列で表示する。

（2） parameter

グラフ表示を行なうバイナリデータを指定する。ファイル名は実際に存在する xxx_cpu 番号というファイル名から_cpu 番号を取り除いた部分で指定する。

表 5-22 全統計情報表示

項目	説明
オプション名	全統計情報表示
文法	ALL
説明	統計情報を全て（呼出回数、平均実行時間、トータル実行時間）をグラフ表示する。

表 5-23 呼出回数表示

項目	説明
オプション名	呼出回数表示（呼出回数 top20）
文法	CNT
説明	呼出回数のグラフを表示する。呼出回数が多いシステムコール 20 個の表示を行なう。

表 5-24 呼出回数、平均実行時間表示

項目	説明
オプション名	呼出回数、平均実行時間表示（呼出回数 top20）
文法	CNT+
説明	呼出回数、平均実行時間のグラフを表示する。呼出回数が多いシステムコール 20 個の表示を行なう。

表 5-25 全統計情報表示（呼出回数 top20）

項目	説明
オプション名	全統計情報表示（呼出回数 top20）
文法	CNT++
説明	呼出回数、平均実行時間、トータル実行時間のグラフを表示する。呼出回数が多いシステムコール 20 個の表示を行なう。

表 5-26 平均実行時間表示

項目	説明
オプション名	平均実行時間表示（平均実行時間 top20）
文法	AVE
説明	平均実行時間のグラフを表示する。平均実行時間が大きいシステムコール 20 個の表示を行なう。

表 5-27 平均実行時間、呼出回数表示

項目	説明
オプション名	平均実行時間、呼出回数表示（平均実行時間 top20）
文法	AVE+
説明	平均実行時間と呼出回数のグラフを表示する。平均実行時間が大きいシステムコール 20 個の表示を行なう。

表 5-28 全統計情報表示（平均実行時間 top20）

項目	説明
オプション名	全統計情報表示（平均実行時間 top20）
文法	AVE++
説明	平均実行時間、呼出回数、トータル実行時間のグラフを表示する。平均実行時間が大きいシステムコール 20 個の表示を行なう。

表 5-29 トータル実行時間表示

項目	説明
オプション名	トータル実行時間表示（トータル実行時間 top20）
文法	TOTAL
説明	トータル実行時間のグラフを表示する。トータル実行時間が大きいシステムコール 20 個の表示を行なう。

表 5-30 トータル実行時間、呼出回数表示

項目	説明
オプション名	トータル実行時間、呼出回数表示（トータル実行時間 top20）
文法	TOTAL+
説明	トータル実行時間と呼出回数のグラフを表示する。トータル実行時間が大きいシステムコール 20 個の表示を行なう。

表 5-31 全統計情報表示（トータル実行時間 top20）

項目	説明
オプション名	全統計情報表示（トータル実行時間 top20）
文法	TOTAL++
説明	トータル実行時間、呼出回数、平均実行時間のグラフを表示する。トータル実行時間が大きいシステムコール 20 個の表示を行なう。

表 5-32 時系列表示

項目	説明
オプション名	時系列表示
文法	EXEC
説明	記録したシステムコール全てについて発行時間と処理時間のグラフ表示を行なう。横軸が発行時刻、縦軸が処理時間となる。

表 5-33 時系列表示（呼出回数 top20）

項目	説明
オプション名	時系列表示（呼出回数 top20）
文法	EXEC-C
説明	記録したシステムコール中の呼出回数の多いもの 20 個について、発酵時間と処理時間のグラフを表示する。横軸が発行時刻、縦軸が処理時間となる。

5.5. 簡易操作用スクリプト

5.5.1. コマンドライン文法

kstrax-simple [option] <parameter>

5.5.2. 概要

kstrax-simple コマンドは、以下の 2 つの機能を提供する

（1） トレースの開始（ファイルへ出力）

ctr_mod モジュール、kstrax_buf モジュール、（kstrax_stub モジュール）をロード後、kstrax コマンドを起動してファイルへ出力する。カーネル空間のバッファサイズや出力ファイル名はデフォルトの設定となる。

（2） トレースの終了

kstrax コマンドを終了後、kstrax_buf モジュールをアンロードする。

一度終了した記録を再開する場合には、（1）の制御を実施するが、記録用のカーネルバッファは初期化されるので、以前のデータはすべて消えていることに注意。

5.5.3. 詳細仕様

（1） option

(a) start：トレースを開始する。（バイナリデータの出力開始）

(b) stop：トレースを終了する。（バイナリデータの出力終了）

（2） parameter

トレース（バイナリ出力）を行なう時間を設定する。オプションで start を選択した場合のみ有効。設定された場合、指定時間が経過すると自動的に終了する。逆に指定されなかった場合は、stop オプションで終了する必要がある。

表 5-34 トレース開始（簡易操作用スクリプト）

項目	説明
オプション名	トレース開始
文法	start <parameter>
説明	カーネル空間で記録を開始し、バイナリファイルへ出力する。parameter は出力する時間を設定する。指定が無い場合は stop オプションで停止する必要がある。 （モジュールロード＋コマンド実行）

表 5-35 トレース終了（簡易操作用スクリプト）

項目	説明
オプション名	トレース終了
文法	stop
説明	バイナリファイル出力を停止し、カーネル空間での記録を終了する。 （コマンド停止＋モジュールアンロード）

5.6. kstrax モジュールのバージョン番号参照コマンド

kstrax モジュールのバージョン番号の参照には Linux のコマンドである dmesg を利用する。

5.6.1. コマンドライン文法

dmesg

5.6.2. 使用説明

本コマンドにて kstrax モジュールの状態を確認することができる。詳細は６. **付録：kstrax の使用例**を参照。

6. 付録：kstrax の使用例

6.1. カーネル空間での記録の開始

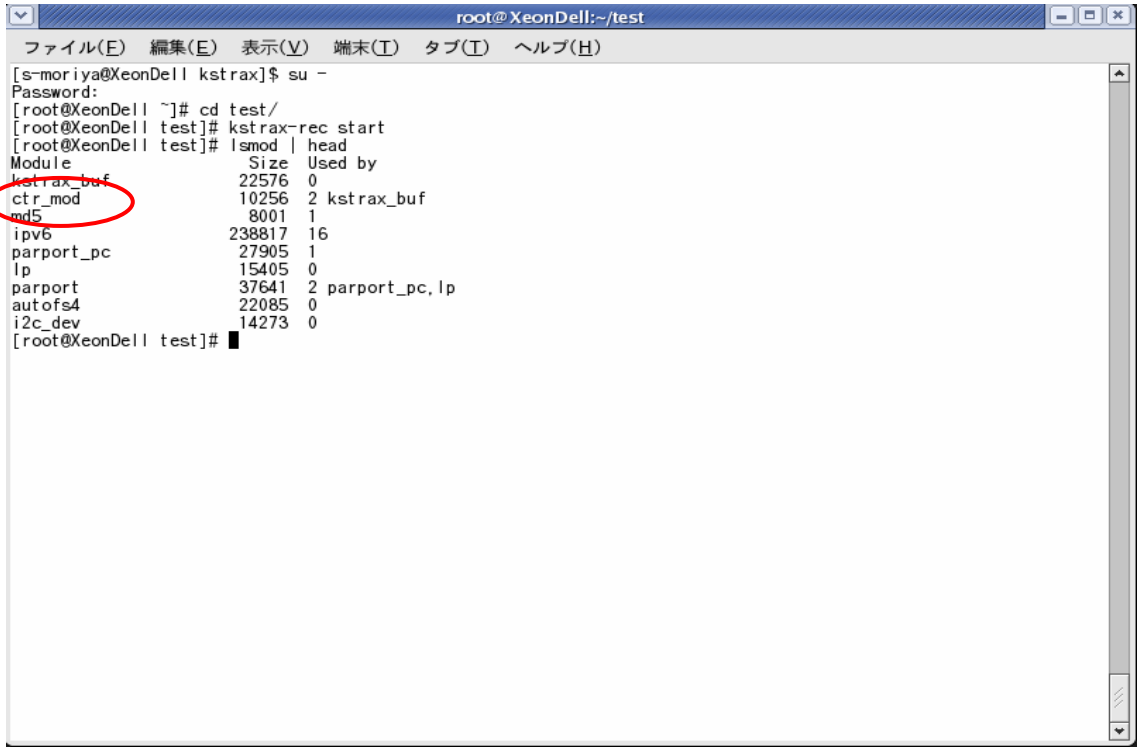


図 6-1 カーネル空間での記録の開始

6.2. バイナリファイルへ出力―①、バイナリファイルを整形してファイルへ出力―②

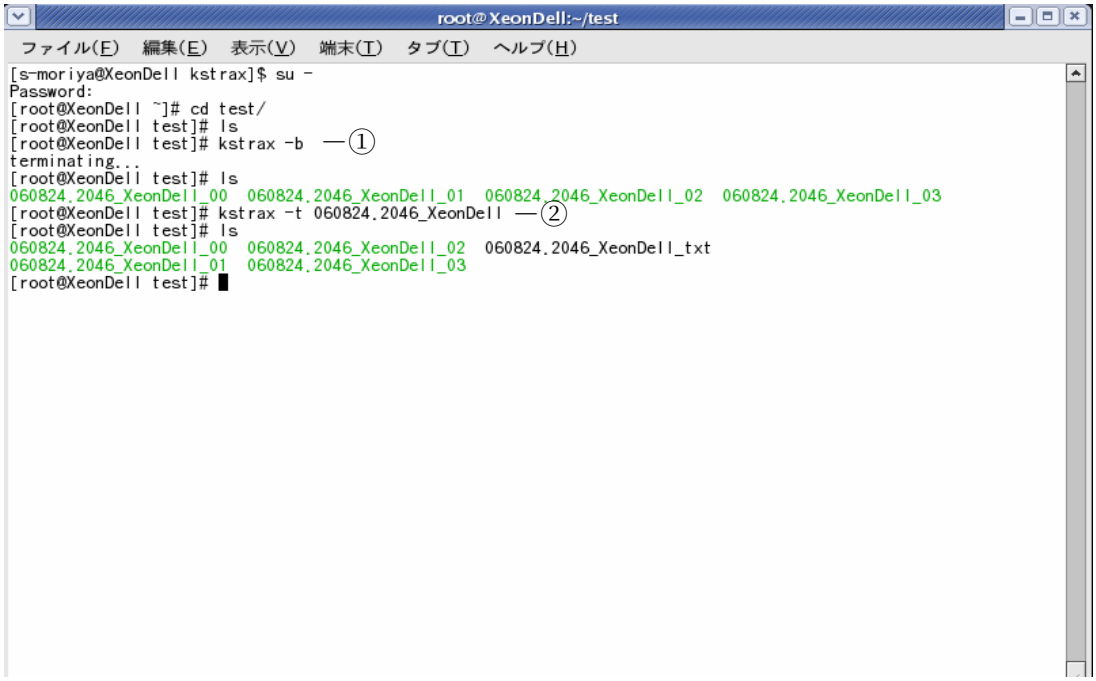


図 6-2 バイナリファイルへ出力、バイナリファイルを整形してファイルへ出力

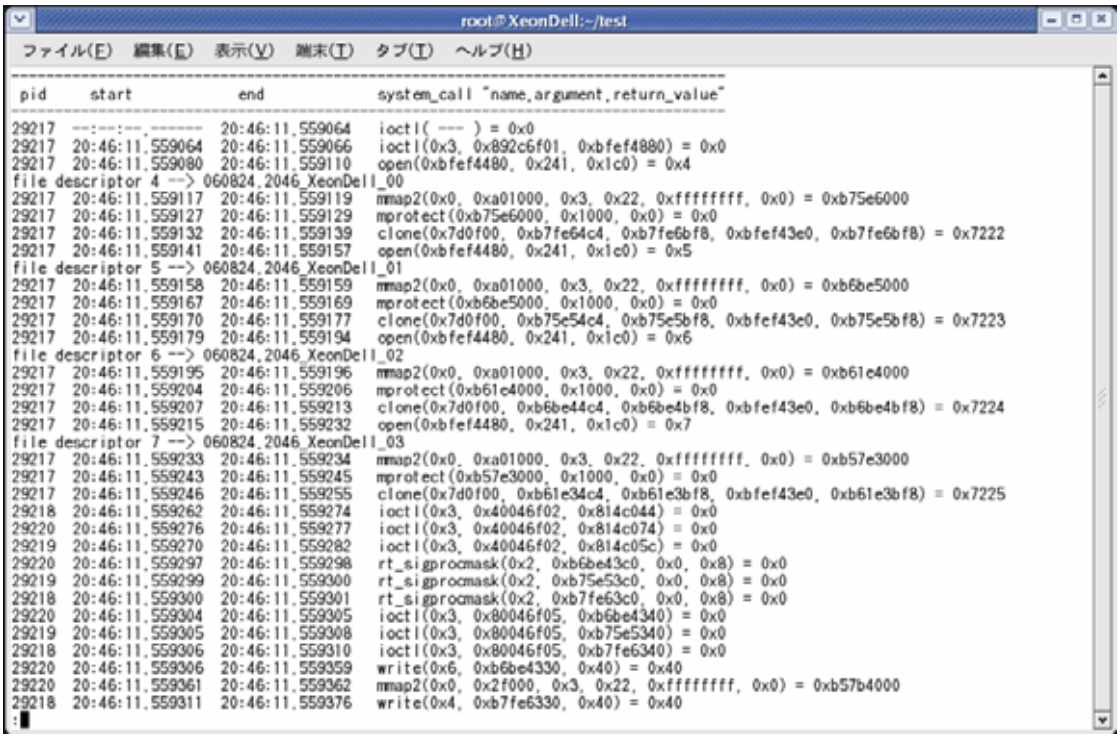


図 6-3 整形後の出力

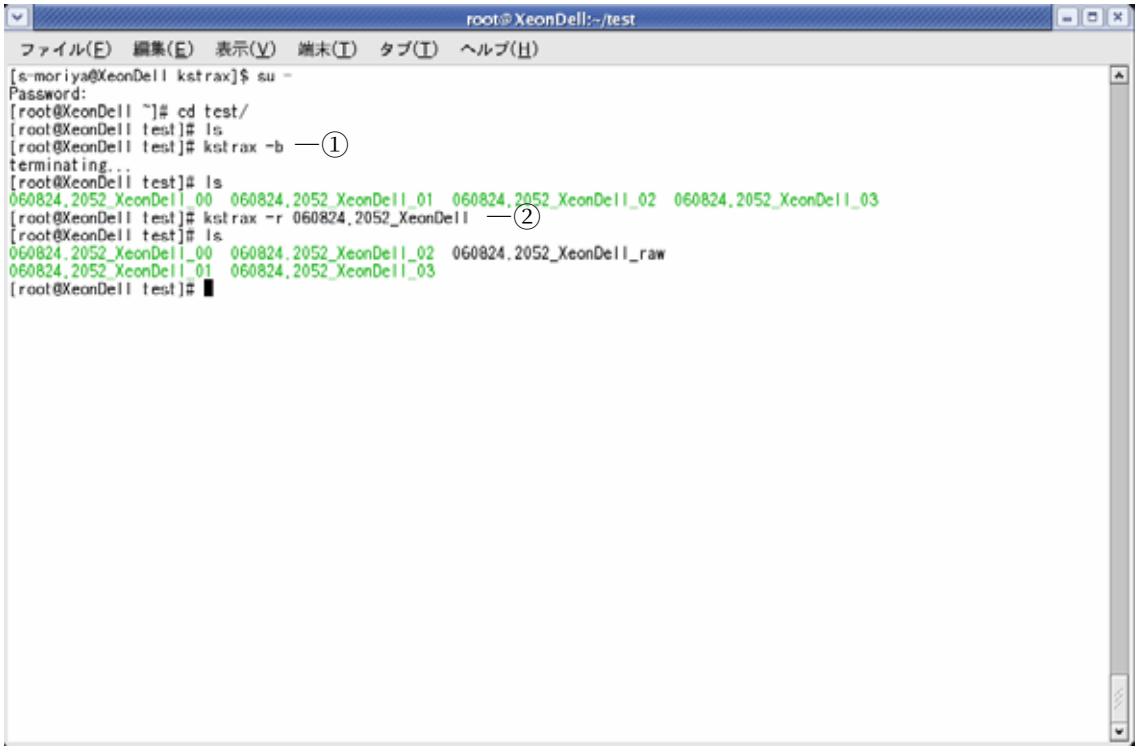


図 6-4 バイナリファイルへ出力、バイナリファイルを整形してファイルへ出力（RAW モード）

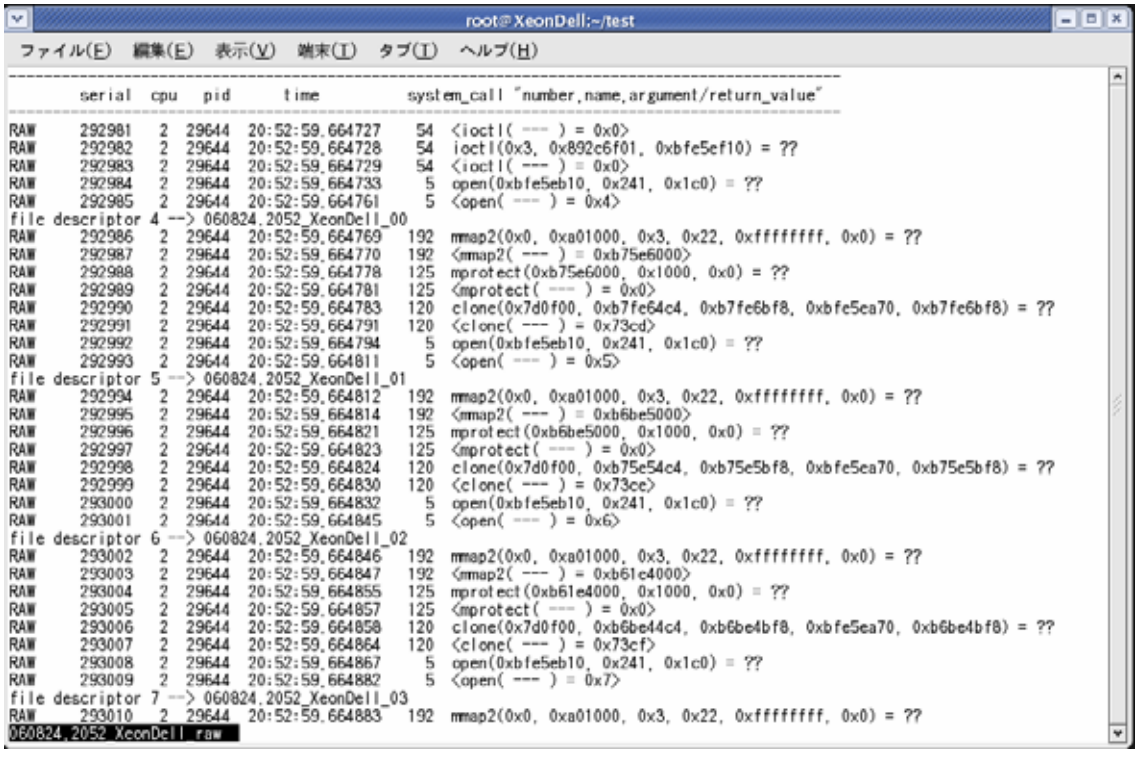


図 6-5 整形後の出力（RAW モード）

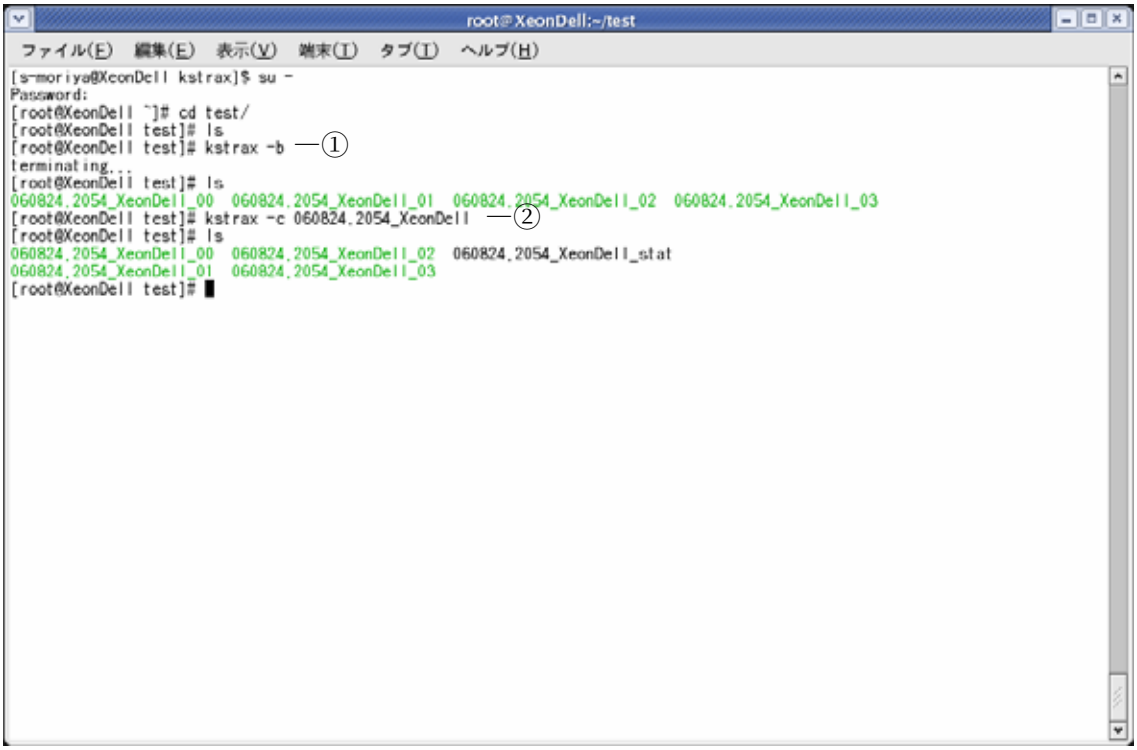
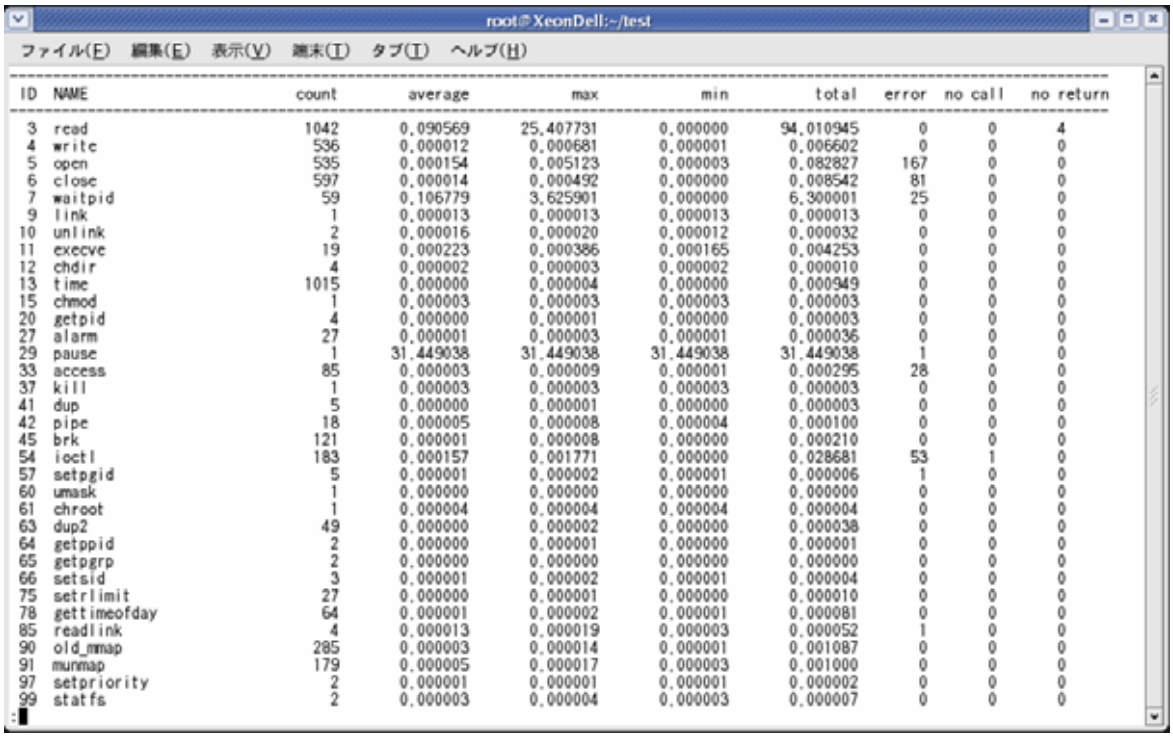


図 6-6 バイナリファイルへ出力、バイナリファイルより統計情報をファイルへ出力



ID	NAME	count	average	max	min	total	error	no call	no return
3	read	1042	0.090569	25.407731	0.000000	94.010945	0	0	4
4	write	536	0.000012	0.000681	0.000001	0.006602	0	0	0
5	open	535	0.000154	0.005123	0.000003	0.082827	167	0	0
6	close	597	0.000014	0.000492	0.000000	0.008542	81	0	0
7	waitpid	59	0.106779	3.625901	0.000000	6.300001	25	0	0
9	link	1	0.000013	0.000013	0.000013	0.000013	0	0	0
10	unlink	2	0.000016	0.000020	0.000012	0.000032	0	0	0
11	execve	19	0.000223	0.000386	0.000165	0.004253	0	0	0
12	chdir	4	0.000002	0.000003	0.000002	0.000010	0	0	0
13	time	1015	0.000000	0.000004	0.000000	0.000949	0	0	0
15	chmod	1	0.000003	0.000003	0.000003	0.000003	0	0	0
20	getpid	4	0.000000	0.000001	0.000000	0.000003	0	0	0
27	alarm	27	0.000001	0.000003	0.000001	0.000036	0	0	0
29	pause	1	31.449038	31.449038	31.449038	31.449038	1	0	0
33	access	85	0.000003	0.000009	0.000001	0.000295	28	0	0
37	kill	1	0.000003	0.000003	0.000003	0.000003	0	0	0
41	dup	5	0.000000	0.000001	0.000000	0.000003	0	0	0
42	pipe	18	0.000005	0.000008	0.000004	0.000100	0	0	0
45	brk	121	0.000001	0.000008	0.000000	0.000210	0	0	0
54	ioctl	183	0.000157	0.001771	0.000000	0.028681	53	1	0
57	setpgid	5	0.000001	0.000002	0.000001	0.000006	1	0	0
60	umask	1	0.000000	0.000000	0.000000	0.000000	0	0	0
61	chroot	1	0.000004	0.000004	0.000004	0.000004	0	0	0
63	dup2	49	0.000000	0.000002	0.000000	0.000038	0	0	0
64	getppid	2	0.000000	0.000001	0.000000	0.000001	0	0	0
65	getpgrp	2	0.000000	0.000000	0.000000	0.000000	0	0	0
66	setsid	3	0.000001	0.000002	0.000001	0.000004	0	0	0
75	setrlimit	27	0.000000	0.000001	0.000000	0.000010	0	0	0
78	gettimeofday	64	0.000001	0.000002	0.000001	0.000081	0	0	0
85	readlink	4	0.000013	0.000019	0.000003	0.000052	1	0	0
90	old_mmap	285	0.000003	0.000014	0.000001	0.001087	0	0	0
91	munmap	179	0.000005	0.000017	0.000003	0.001000	0	0	0
97	setpriority	2	0.000001	0.000001	0.000001	0.000002	0	0	0
99	statfs	2	0.000003	0.000004	0.000003	0.000007	0	0	0

図 6-7 統計情報出力

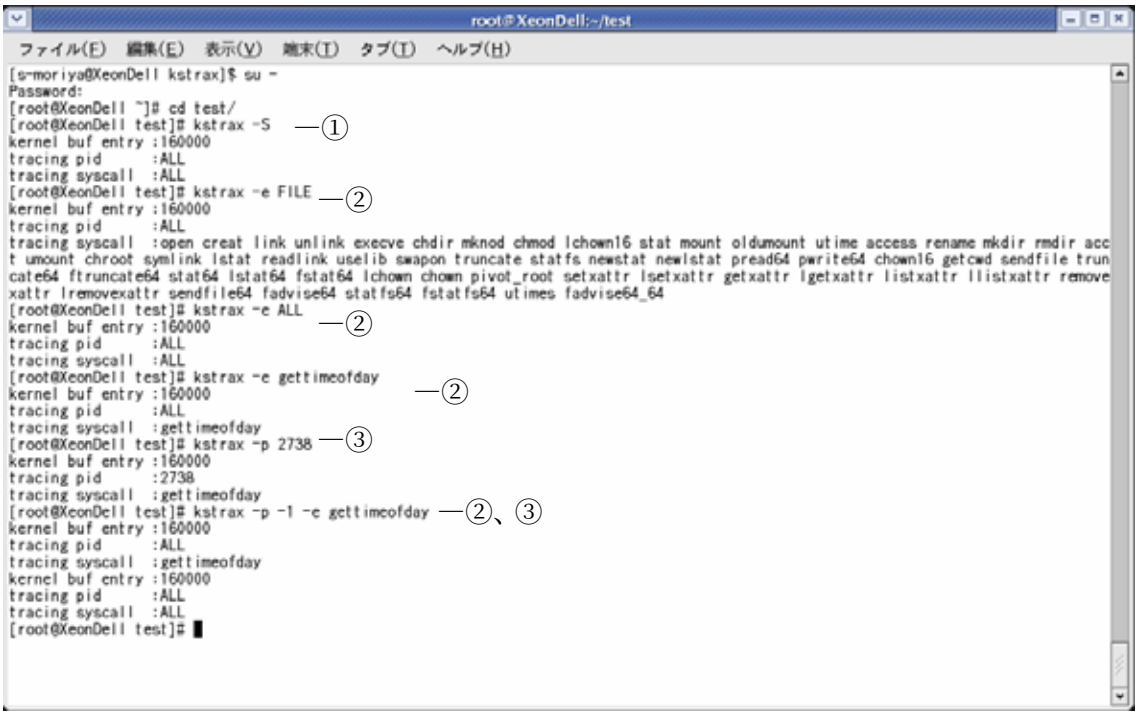


図 6-8 トレース情報の確認①、記録するシステムコールの種類②、プロセスの指定③

6.3. カーネル空間での記録終了

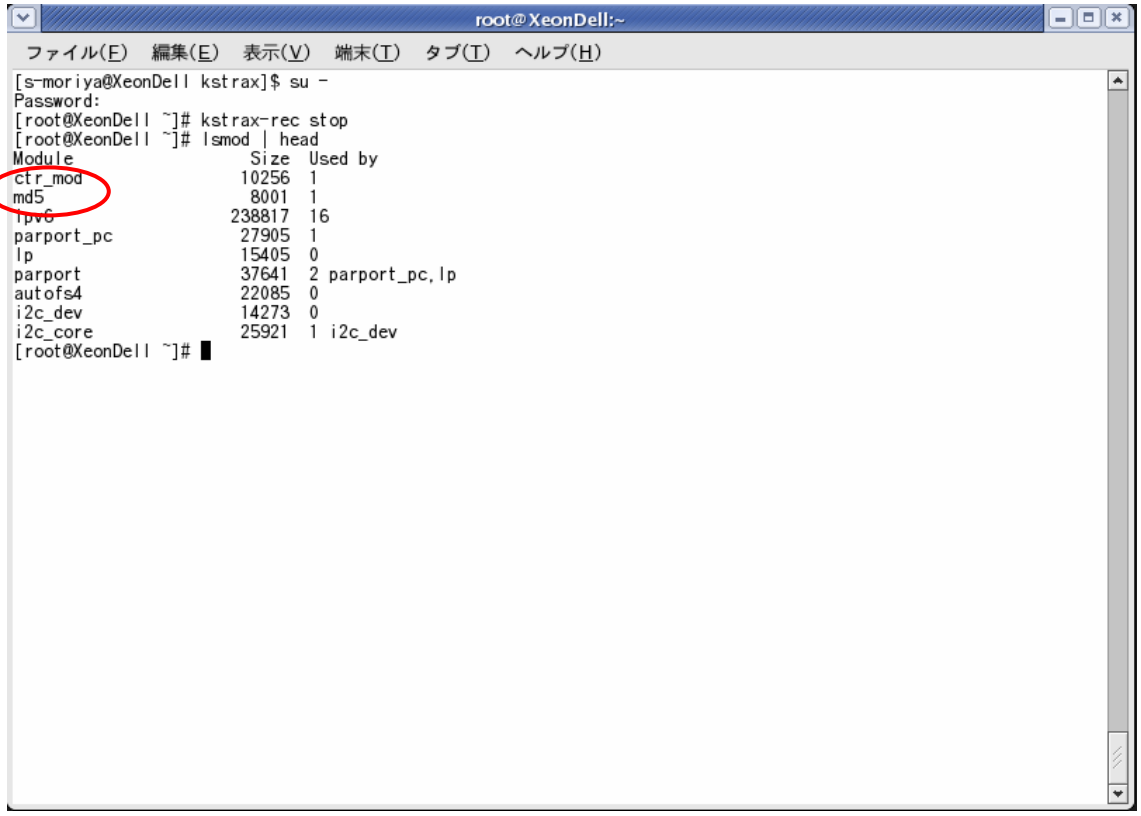


図 6-9 カーネル空間での記録終了

6.4. 採取したデータをフィルタリングして表示

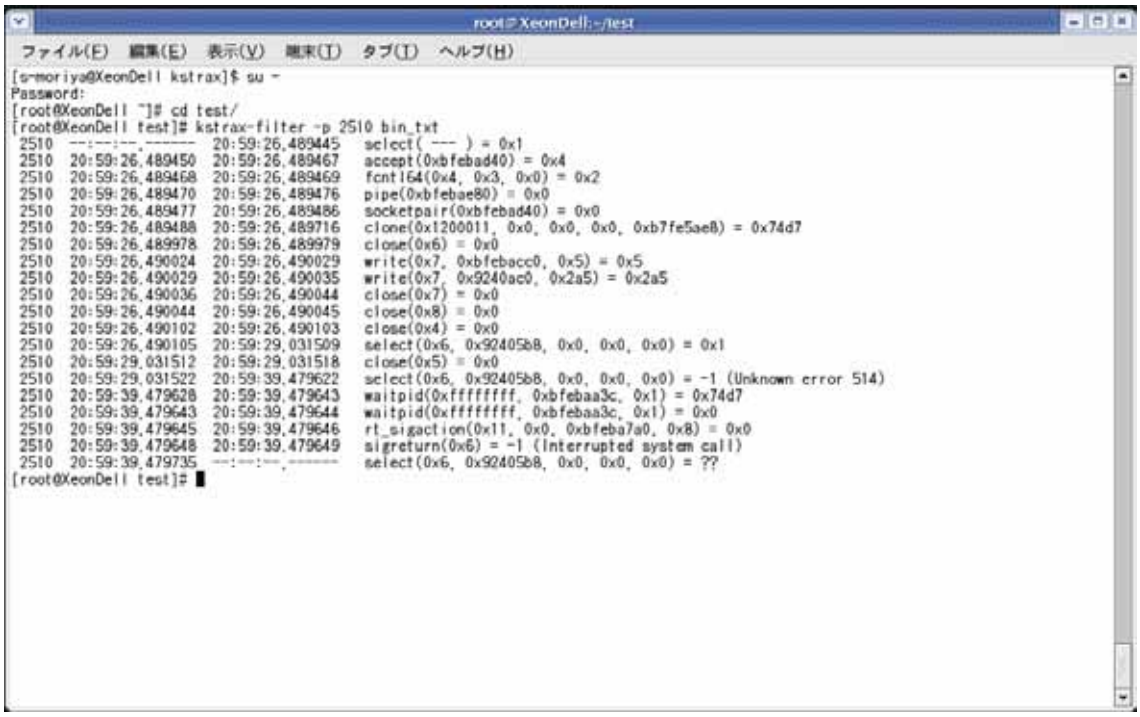


図 6-10 PID でフィルタリング

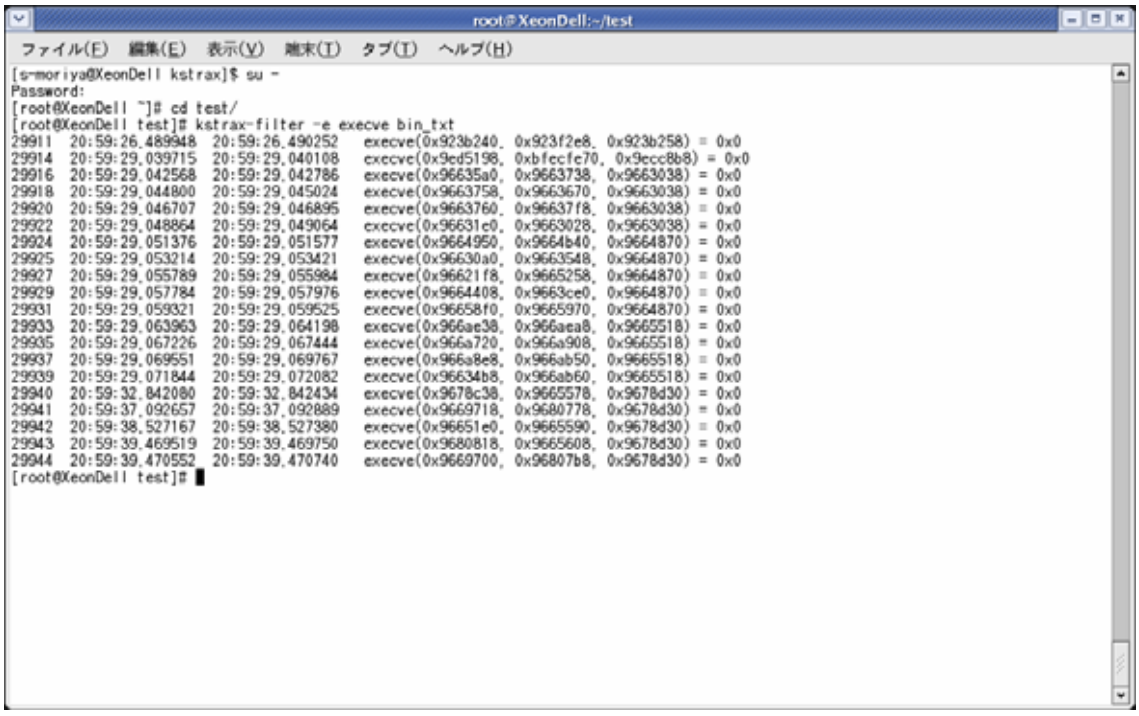


図 6-11 システムコール名でフィルタリング

6.5. グラフ表示

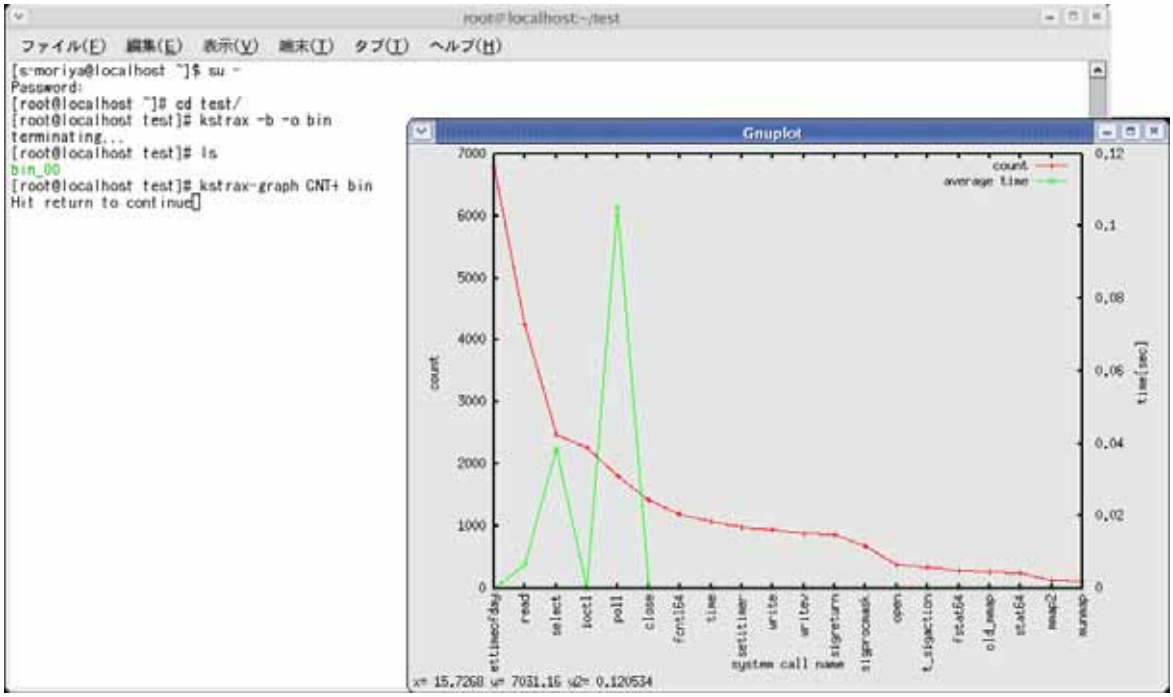


図 6-12 グラフ表示（呼出回数、平均実行時間表示）

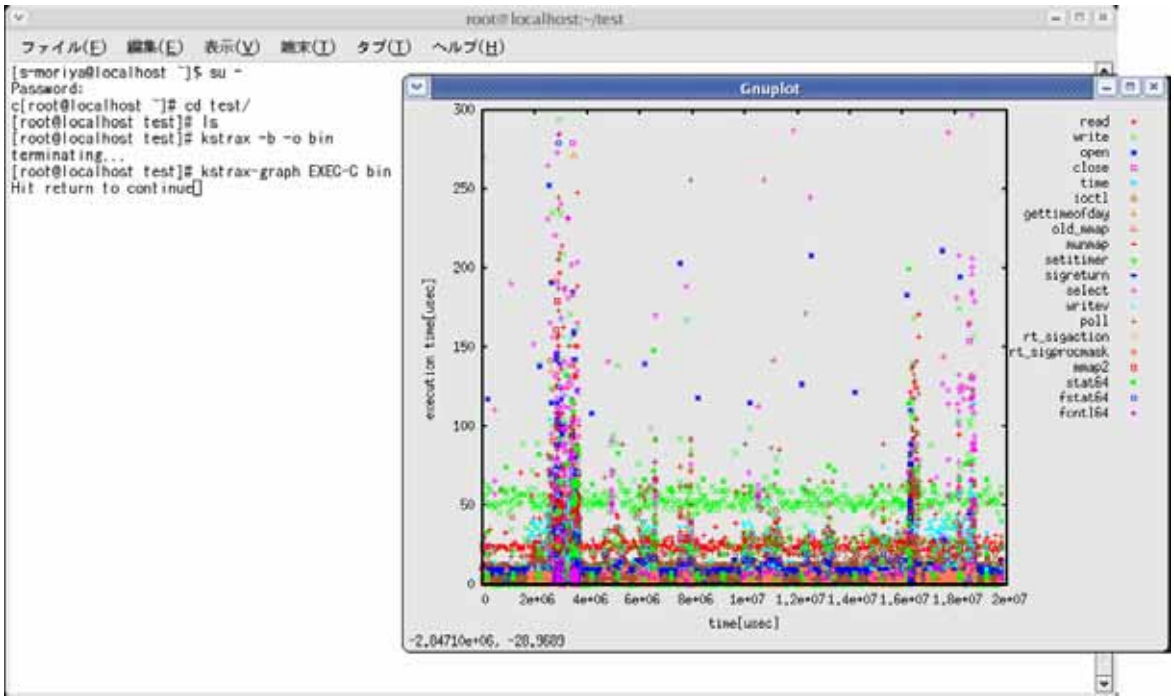


図 6-13 グラフ表示（時系列表示（呼出回数 top20））