平成 28 年度 卒業論文概要			
所　　属	機械情報工学科	指導教員	光來 健一
学生番号	13237029	学生氏名	小野　静流
論文題目	実行時情報に基づく OS カーネルのコンフィグ最小化		

1　はじめに

OS カーネルは OS の中核を構成するソフトウェアであり，主な役割として CPU やメモリなどのハードウェアの資源管理や，アプリケーションの制御・管理がある．OS カーネルの規模は年々大きくなっており，Linux 4.4 のソースコードは約 1 GB にもなる．これは OS カーネルが数多くのデバイスドライバを含む様々な機能を提供しているためである．標準の OS カーネルでは様々なユーザの要求を満たすために非常に多くの機能が有効になっているが，個々のユーザにとってはその多くが不要な機能である．そのため，OS カーネルをカスタマイズすることで，セキュリティや性能を向上させたり，メモリやディスクの消費量を抑えたりすることが必要になる．しかし，OS カーネルのコンフィグ項目は非常に多く，カスタマイズには専門知識が必要となるため，一般のユーザが行うのは難しい．

本研究では，OS カーネルの実行時の情報に基づいて不要な機能を自動的に見つけ，その機能に対応するコンフィグ項目を無効化することでコンフィグの最小化を行う config-mini を提案する．

2　OS カーネルのカスタマイズ

OS カーネルは多様なユーザの要求を満たすために様々な機能を提供している．例えば，多様な実行環境に対応するために膨大な数のデバイスドライバが用意されており，様々なファイルシステムやネットワークプロトコルがサポートされている．それらの機能の多くはカーネルモジュールとして OS カーネル本体とは別に提供され，必要な時にロードされるようになっている．Ubuntu 16.04 の標準カーネルではカーネルモジュールの数は 4,000 以上あり，それ以外にも OS カーネルに組み込まれている機能が 2,000 以上ある．すべてのユーザにとってこれらの機能すべてが必要となることはなく，提供されている機能の多くは不要である．

そのため，OS カーネルは個々のユーザごとにカスタマイズして利用するのが望ましい．第一に，OS カーネルの機能が増えるほど脆弱性も増える傾向にあるため，必要最小限の機能に抑えることでシステムのセキュリティを向上させることができる．第二に，OS カーネルが提供する機能の中にはシステムの性能を低下させるものもあるため，そのような機能が不要であれば無効化することによって性能を向上させることができる．第三に，OS カーネルの機能を減らせばその分だけサイズを小さくすることができるため，メモリやディスクの消費量を抑えることができる．これは小さな組み込み機器においては重要である．

OS カーネルのカスタマイズは以下の手順で行うことができる．まず，OS カーネルのコンフィグを変更することで，不要な機能を無効化する．OS カーネルの各機能にはそれぞれコンフィグ項目が用意されており，必要に応じてコンフィグ項目を設定することで OS カーネルに含める機能を選択することができる．次に，変更したコンフィグを用いて OS カーネルをコンパイルし，システムにインストールする．その後で，システムを再起動して動作確認を行う．コンフィグで無効化した機能が利用できなくなっており，かつ，システムの動作に支障がないことが確認できればカスタマイズの完了となる．

しかし，OS カーネルのカスタマイズには専門知識が必要であり，一般のユーザが行うのは難しい．OS カーネルのコンフィグを変更する際に，OS カーネルやコンフィグに対する知識がなければどの機能が不要なのかを判断することができない．また，不要な機能が見つかったとしても，8,000 以上のコンフィグ項目の中からその機能に対応しているコンフィグ項目を見つけることも困難である．そのため，コンフィグを少し変更して OS カーネルをコンパイルし直し，動作確認を行うという試行錯誤が必要となる．コンフィグの変更の仕方によってはシステムが起動しなくなる恐れもある．また，OS カーネルのコンパイルやシステムの再起動には時間がかかるため，試行錯誤を繰り返すには膨大な時間が必要となる．

3　config-mini

本研究では，OS カーネルの実行時情報に基づいて不要な機能を自動的に見つけ，その機能に対応するコンフィグ項目を無効化することでコンフィグを最小化を行う config-mini を提案する．OS カーネルが実際にどの機能を使っているかを判断するために，config-mini はカーネル関数の実行ログを取得し，ある機能を構成する関数が実行されていない場合に，その機能に対応するコンフィグ項目を無効化する．

3.1　OS カーネルの実行ログの取得

config-mini はそれぞれの機能ごとに必ず実行される関数のみの実行ログを取得する．これは，OS カーネルの実行ログをすべての関数について取得するのは，カーネル関数の数が膨大であることから現実的ではないためである．機能の初期化を行う際に使われる関数は必ず実行されるが，初期化が行われたとしてもその機能が実際に「使われる」とは限らない．例えば，OS カーネルに組み込まれているシリアルデバイスのドライバは必ず初期化されるが，シリアルポートを用いた通信を行わなければその機能は不要とみなせる．そこで，機能が「使われた」と判断できるキーとなる関数を実行ログとして取得する．例えばシリアルドライバの場合には，シリアルポートを使った場合と使わなかった場合の実行ログを比較して，`serial8250_get_mctrl` をキーとなる関数とした．

カーネル関数の実行ログは Linux の ftrace 機構を利用して

取得する．ftrace は OS カーネルに標準で組み込まれているトレース機構であり，その中の関数トレーサを用いて実行された関数を記録する．キーとなる関数だけの実行ログを取得するために，ftrace のフィルタ機能を用いて OS カーネルの起動オプションでログを取得する関数を限定する．しかし，ftrace はフィルタを設定する時点，つまり，OS カーネルの起動開始時に存在する関数のみしか指定できないため，OS カーネルの起動中や起動後にロードされたカーネルモジュール内の関数については実行ログを取得することができない．そこで，カーネルモジュールローダを修正し，カーネルモジュールがロードされるたびに ftrace のフィルタを再設定するように変更することで，カーネルモジュールの関数のログも取得できるようにした．

3.2 関数に対応するコンフィグの特定

キーとなる関数が実行されなかった機能は不要な機能であると判断し，config-mini はその機能がどのコンフィグ項目に対応しているかを特定する．まず，キーとなる関数が OS カーネルのソースコードの中のどのファイルで定義されているのかを特定する．関数とファイルを対応づけるために，gtags を用いてタグファイルと呼ばれるデータベースを作成する．gtags はソースコードを解析し，関数などの情報をタグファイルに格納する．このタグファイルを用いて global コマンドを実行することで，関数が定義されているファイルのパス名を取得することができる．例えば，シリアルドライバのキーとなる関数が定義されているファイルのパス名は `drivers/tty/serial/8250/8250_port.c` であるとわかる．

次に，config-mini は取得したファイルに対応するコンフィグ項目を特定する．そのために，OS カーネルのソースコードに含まれている Makefile を解析して，データベースを作成する．Makefile にはどのコンフィグが有効な時にどのファイルまたはディレクトリをコンパイルするかというルールが下記のように書かれている．

```
obj-$(CONFIG_SMP) += smp.o
obj-$(CONFIG_KVM) += kvm/
```

作成したデータベースを用いて，ファイルのパス名に一致するコンフィグ項目を取得する．パス名に一致する項目が見つからない場合，パス名に含まれるディレクトリ名が最も長く一致するコンフィグ項目を取得する．例えば，`drivers/tty/serial/8250/8250_port.c` に対応するコンフィグ項目は見つからないため，`drivers/tty/serial/8250/`のようにパス名を短くしていき対応するコンフィグ項目を見つける．シリアルドライバの場合には最終的に `CONFIG_SERIAL_8250` というコンフィグ項目が見つかる．

config-mini はサポートしている機能すべてについて以上の工程を自動的に実行する．ユーザはカーネル関数の実行ログから不要と判断された機能に対応するキーとなる関数をファイルに記述しておく．config-mini はファイル内の関数を 1 つずつ読み取り，それぞれの関数に対応するコンフィグ項目を見つける．そして，現在のコンフィグから不要な項目を無効化したコンフィグを生成する．有効になっているコンフィグ項目を無効化するには，その項目を消すのではなく，

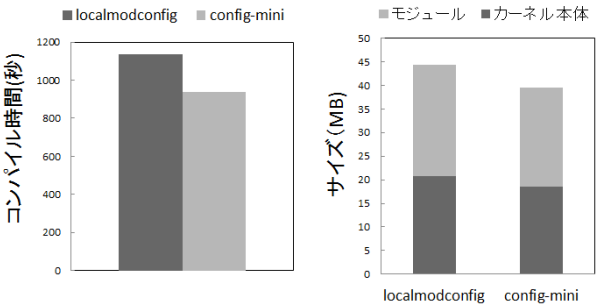

図 1　OS カーネルのコンパイル時間とサイズ

```
# CONFIG_SERIAL_8250 is not set
```

のように変更する．

4 実験

config-mini によりいくつかの不要な機能を減らしたコンフィグを用いて，OS カーネルをコンパイルしたときのコンパイル時間とサイズを調べる実験を行った．実験には，Intel Xeon E3-1226 v3 の CPU，8GB のメモリ，457GB の HDD を備えたマシンを用いた．Windows 10 上で VirtualBox 5.0.20 を動作させ，仮想マシン内で対象 OS の Linux 4.4.0 を動作させた．まず，CD ドライバのキーとなる関数である `isofs_readpages` 関数からその機能が対応しているコンフィグ項目を自動で見つけられることを確かめた．config-mini を用いることにより，この関数は `CONFIG_ISO9660_FS` というコンフィグ項目に対応することが分かった．その項目を無効化したコンフィグを用いて Linux を再コンパイルして再起動したところ，CD ドライバに関するカーネルモジュールがロードされなくなることを確認した．

次に，Linux の localmodconfig 機能を用いて生成されたコンフィグと，それに対して config-mini を適用したものを用いて，OS カーネルのコンパイル時間とサイズを測定した．localmodconfig はカーネルモジュールになっている機能のうち，実行時にロードされたものに対応する項目のみを有効にしたコンフィグを生成する．ただし，config-mini とは異なり，機能の初期化を行っただけでも必要な機能とみなされる．図 1 に実験結果を示す．config-mini で生成したコンフィグの方がコンパイル時間は 195 秒短くなり，サイズは 4.8MB 小さくなった．

5 まとめ

本研究では，OS カーネルの実行時情報に基づいて不要な機能を自動的に見つけ，その機能に対応するコンフィグ項目を無効化することでコンフィグを最小化する config-mini を提案した．現在のところ，利用可能ないくつかの機能に対してキーとなる関数を見つけることができているが，利用できない機能については見つけられていない．様々な実行環境を用意してキーとなる関数を見つけることが今後の課題である．