

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第3824580号  
(P3824580)

(45) 発行日 平成18年9月20日(2006.9.20)

(24) 登録日 平成18年7月7日(2006.7.7)

(51) Int. Cl.		F I		
<b>G06F</b>	<b>9/445</b>	<b>(2006.01)</b>	G06F	9/06 640A
<b>G06F</b>	<b>21/22</b>	<b>(2006.01)</b>	G06F	9/06 660A
<b>G06F</b>	<b>11/10</b>	<b>(2006.01)</b>	G06F	11/10 310B

請求項の数 2 (全 15 頁)

<p>(21) 出願番号 特願2002-372672 (P2002-372672)</p> <p>(22) 出願日 平成14年12月24日 (2002.12.24)</p> <p>(65) 公開番号 特開2003-223329 (P2003-223329A)</p> <p>(43) 公開日 平成15年8月8日 (2003.8.8)</p> <p>審査請求日 平成14年12月24日 (2002.12.24)</p> <p>(31) 優先権主張番号 2001-86661</p> <p>(32) 優先日 平成13年12月28日 (2001.12.28)</p> <p>(33) 優先権主張国 韓国 (KR)</p>	<p>(73) 特許権者 500562019                  エス ケイ テレテック カンパニー リ                  ミティッド                  大韓民国ソウル市江南区駅三洞737番地                  スター・タワー21階                  21st floor, star Tow                  er, 737, Yeoksam-dong                  kangnam-gu, Seoul 1                  35-984, Republic of                  Korea</p> <p>(74) 代理人 100109416                  弁理士 鈴木 健治</p>
---	--

最終頁に続く

(54) 【発明の名称】 移動通信携帯端末及びダウンロードプログラムファイル管理用プログラム

(57) 【特許請求の範囲】

【請求項1】

プログラムファイル及びデータを記憶するメモリと、このメモリを使用してプログラムファイルを実行するプロセッサと、このプロセッサ又は論理回路の制御に従ってファイルのダウンロード等の通信を制御する無線通信部とを備えた移動通信端末であって、

前記プロセッサで直接実行できる機械語コードと、少なくとも当該機械語コードのマジックコード、チェックサム及びアプリケーションIDとを格納するヘッダとを備えた特定のデータ構造のプログラムファイルをダウンロードすると共に、前記メモリに格納するダウンロード制御部と、

実行指令が入力された時に、前記メモリに格納されたプログラムファイルを当該プログラムファイルのヘッダに含まれた各データを用いて当該プログラムファイルの有効性を検証する有効性検証機能と、

前記有効性が検証された場合に限り、当該プログラムファイルの機械語コードを前記メモリにロードし或いは前記プロセッサが直接にアクセスして、当該プログラムファイルの機械語コードを実行する実行部とを備え、

前記チェックサムが、前記機械語コードのチェックサム値であって、

前記マジックコードが、前記実行部の論理演算命令を使用した場合と同一の論理演算結果となる論理演算を用いて、前記アプリケーションIDとチェックサムとに論理演算をした論理演算値であって、

前記有効性検証機能が、前記実行指令が入力された時に、当該プログラムファイルのヘ

10

20

ッダの前記アプリケーションIDと前記チェックサムとに前記実行部の論理演算をし、前記プログラムファイルのヘッダのマジックコードと当該論理演算結果とを比較して、一致した際に有効性有りとし、

前記実行部が、前記有効性有りとされたプログラムファイルの機械語コードを実行することを特徴とする移動通信端末。

**【請求項2】**

プログラムファイル及びデータを記憶するメモリと、このメモリを使用してプログラムファイルを実行するプロセッサと、このプロセッサ又は論理回路の制御に従ってファイルのダウンロード等の通信を制御する無線通信部とを備えた移動通信端末を使用して前記プログラムファイルを前記プロセッサに実行させるためのダウンロードプログラムファイル管理用プログラムであって、

前記チェックサムが、前記機械語コードのチェックサム値であって、

前記マジックコードが、前記実行部の論理演算命令を使用した場合と同一の論理演算結果となる論理演算を用いて、前記アプリケーションIDとチェックサムとに論理演算をした論理演算値であって、

前記移動通信端末用プログラムは、前記移動通信端末を動作させる指令として、

前記プロセッサで直接実行できる機械語コードと、少なくとも当該機械語コードのマジックコード、チェックサム及びアプリケーションIDとを格納するヘッダとを有する特定のデータ構造のプログラムファイルをダウンロードすると共に、前記メモリに格納する第1指令と、

前記プログラムファイルのヘッダに含まれた前記各データを用いて当該プログラムファイルの有効性を検証させる第2指令と、

前記第2指令に応じて有効性が検証される場合に限って、当該プログラムファイルの機械語コードを前記メモリにロードし或いは前記プロセッサが直接にアクセスして、当該プログラムファイルの機械語コードを実行させる第3指令とを備え、

前記第2指令が、前記プログラムファイルのヘッダの前記アプリケーションIDと前記チェックサムとに前記プロセッサ又は論理回路の論理演算をし、前記プログラムファイルのヘッダのマジックコードと当該論理演算結果とを比較して、一致した際に有効性有りとなさせ、

前記第3指令が、前記有効性有りとされたプログラムファイルの機械語コードを実行させることを特徴とするダウンロードプログラムファイル管理用プログラム。

**【発明の詳細な説明】**

**【0001】**

**【発明の属する技術分野】**

本発明は、機械語コードからなる実行領域を含む移動通信端末用ファイル構造及びそれを用いたファイル実行方法に関し、さらに詳しくは、端末のプロセッサによって直接実行可能な機械語実行コードを含み、ファイルの信頼性・有効性検証、ファイルの実行可能回数または時間制限、及び自動更新のためのURL情報を含むヘッダを備える移動通信端末用ファイル構造及びそれを用いた実行方法に関する。

**【0002】**

**【従来の技術】**

最近、移動通信サービスの普及に伴って、殆どの人に1つ以上の移動通信サービスが提供されている。このような移動通信サービスは、基本的な音声通話、文字・音声メッセージ伝送、ひいては無線インターネットサービス、移動通信端末用コンテンツの利用などに拡大されつつある。

その中でも、無線インターネットサービスは、端末を用いてインターネット上のサーバーに接続して必要な情報を入手するうえ、各種イメージ、ベル音（着信音、着信メロディー）、ゲームなどの無線インターネット・コンテンツ・ファイルを無線でダウンロードして利用することを可能にする。

**【0003】**

10

20

30

40

50

移動通信端末用ファイルのダウンロード及び実行のために、主に運営体制又はハードウェア独立なプラットフォーム(platform)を利用しており、移動通信端末用ファイルは所定のプログラム開発ツール(Development Kit)を用いて作成する。

移動通信端末用ファイルは、端末メモリ制限などによって小容量にしなければならず、プログラミング及び実行の便宜上、標準化されたライブラリ(Library)のAPIを用いる。このようなAPIは端末のコンテンツ実行用プラットフォームで使用可能であり、コンテンツ用応用プログラム(アプリケーションプログラム)を単純化及び低容量化することができる。

#### 【0004】

ところが、従来の端末用ファイル実行のためには、端末のプロセッサによって直接実行が不可能なバイトコード基盤のプログラムをダウンロードし、端末の実行プラットフォームとして既に内蔵されているKVM、GVM、JVMなどの仮想マシン(VM:Virtual Machine)を用いてバイトコードを解釈し、機械語に変換(翻訳, interpret)した後実行する。JAVA(商標)基盤の無線プラットフォームは前述した方式で実行されるが、このような方式はプラットフォーム独立なので他機種間の互換性は増加させたが、プログラムの実行時間が長くなり、プログラムを格納するためのメモリ空間も多く浪費されるという欠点がある。

10

#### 【0005】

図1は従来の移動通信端末用ファイルの内部構成を示すもので、ファイルの実行領域とソース領域は全く分離されておらず1つの領域内に含まれており、実行領域はバイトコードからなる実行コードと実行データに区分されている。

20

このような、ファイルをダウンロードした端末のプラットフォームは、実行のために仮想マシンを用いてバイトコードを翻訳して機械語に変換した後、実行しなければならない。従って、実行に長い時間がかかるうえ、通常コンパイルされた機械語よりはバイトコードがより大きい容量を有するので、比較的メモリ空間の小さい端末に多くのファイルを格納することができないという欠点があった。

#### 【0006】

また、バイトコードを使用する場合には、使用メモリ容量を予測することが困難で、プログラムを実行するために大量のメモリを使用する。このため、移動通信端末はより大きいメモリを搭載しなければならず、この大きいメモリ容量と、インタプリタによる変換を含むプログラムの長い実行時間とから、プログラムファイルをダウンロードして実行できる移動通信端末は、そのバッテリーの稼働時間が短くなってしまふ。

30

#### 【0007】

通常、プログラムの遠隔ダウンロード及び実行時には、バイトコードで作成されたプログラムをダウンロードして仮想マシンを用いてバイトコードを機械語に翻訳した後実行する方式(JAVA(商標)、C++)が主に用いられている。ところが、このような翻訳実行方式はファイルを実行するプロセッサがそれぞれ異なるために必要なものであって、全てのファイル実行プロセッサが同じ形態であれば、容量が小さくて実行速度も速い機械語からなるファイルを用いることが便利である。

#### 【0008】

40

一方、移動通信端末に使用されるプロセッサは、ハードウェア的又は運営体制OSの面においてほぼ統一した仕様を有しているので、敢えてバイトコードを使用する必要がない。本発明は、かかる点に着目したもので、有効性検証、使用期間制限及び自動更新機能を備えるヘッダと、機械語コードからなる実行領域と、言語別に選択されるリソース領域とからなる移動通信端末用ファイルの構造を提案する。

#### 【0009】

##### 【発明が解決しようとする課題】

本発明の目的は、機械語コードを含むことにより、容量を減少させ且つ実行速度を向上させることが可能な移動通信端末用ファイル構造を提供することにある。

本発明の他の目的は、ヘッダと、機械語コードからなる実行領域と、言語別に選択される

50

リソース領域とからなる移動通信端末用ファイル構造を提供することにある。

【0010】

【課題を解決するための手段】

上記目的を達成するための本発明に係る移動通信端は、プログラムファイル及びデータを記憶するメモリと、このメモリを使用してプログラムファイルを実行するプロセッサと、このプロセッサ又は論理回路の制御に従ってファイルのダウンロード等の通信を制御する無線通信部とを備えている。

そして、前記プロセッサで直接実行できる機械語コードと、少なくとも当該機械語コードのマジックコード、チェックサム及びアプリケーションIDとを格納するヘッダとを備えた特定のデータ構造のプログラムファイルをダウンロードすると共に、前記メモリに格納するダウンロード制御部と、実行指令が入力された時に、前記メモリに格納されたプログラムファイルを当該プログラムファイルのヘッダに含まれた各データを用いて当該プログラムファイルの有効性を検証する有効性検証機能と、前記有効性が検証された場合に限り、当該プログラムファイルの機械語コードを前記メモリにロードし或いは前記プロセッサが直接にアクセスして、当該プログラムファイルの機械語コードを実行する実行部とを備えている。

10

【0011】

そして、前記チェックサムが、前記機械語コードのチェックサム値であって、前記マジックコードが、前記実行部の論理演算命令を使用した場合と同一の論理演算結果となる論理演算を用いて、前記アプリケーションIDとチェックサムとに論理演算をした論理演算値である。

20

【0012】

さらに、前記有効性検証機能が、前記実行指令が入力された時に、当該プログラムファイルのヘッダの前記アプリケーションIDと前記チェックサムとに前記実行部の論理演算をし、前記プログラムファイルのヘッダのマジックコードと当該論理演算結果とを比較して、一致した際に有効性有りとし、前記実行部が、前記有効性有りとされたプログラムファイルの機械語コードを実行する。

【0013】

【発明の実施の形態】

以下、添付図に基づいて本発明の実施形態を詳細に説明する。

30

図2は本発明に係る移動通信端末用ファイルの内部構成を示すもので、まずヘッダ領域と、機械語形態の実行コード及び実行データからなる実行領域と、ファイル実行時に実行領域が参照するデータとしてのリソース領域とからなっている。

実行領域の実行コード及び実行データは、作成されたソースコードまたはバイトコードをコンパイルまたは変換(翻訳)することにより生成される。機械語コードは、0又は1を1ビットとする2進数であり、その一部がプロセッサに対する命令に、残りの部分がメモリのアドレスや変数として用いられる。

【0014】

次いで、ファイルヘッダの詳細な構成を図3、リソース領域の細部構成を図4に基づいてそれぞれ詳細に説明する。

40

図3は本発明に使用されるファイルヘッダの詳細構成図であって、ファイルヘッダは、プロセッサによって直接実行または使用される実行領域及びリソース領域とは異なり、それ自体では使用されないが、特殊機能の実現に必要なデータを含む部分である。

【0015】

本発明に使用されるファイルヘッダは固定フィールド(Fixed Field)と可変フィールド(Variable Field)に大別される。

固定フィールドは、固定の長さを有する領域であって、ファイルを実行するための実行領域の位置及びサイズデータ、実行領域に含まれたプロセッサによる実行開始位置に対する情報、指定された領域及び言語に対応するように含まれたデータの集合であるリソース領域の位置及びサイズ、ファイルの有効性及び信頼性を確認するためのデータなどを含んで

50

いる。

ヘッダの固定フィールドに含まれるデータの種別を具体的に考察すると、下記表 1 の通りである。

【 0 0 1 6 】

【 表 1 】

フィールド名	大きさ	特性または機能	
マジックナンバー (Magic Number)	4	Checksum と application ID を XOR した値。	
チェックサム (Checksum)	4	ファイルヘッダのヘッダサイズから残り領域、実行領域及びリソース領域の全データを 4 バイト単位で加算して、下位 4 バイトのみ取った値。	10
ヘッダサイズ (Header Size)	2	固定領域と可変領域を含むヘッダ全体のバイト数	
アプリケーション ID (Application ID)	4	該当ファイルが具現するアプリケーションの ID。	
アプリバージョン (Application version)	2	該当ファイルが具現するアプリケーションのバージョン。	
タイプ (Type)	2	実行ファイルか或いはライブラリファイルか。API アクセスレベルなどを定めるビット・フィールド。	
プラットフォームバージョン (Platform version)	4	本ファイルの実行領域機械語コードを実行させる実行プラットフォームのバージョン。	20
ROセグメントサイズ (RO segment Size)	4	実行領域に含まれた機械語実行コード部分 (RO segment) のサイズ。	
RWセグメントサイズ (RW Segment Size)	4	実行領域に含まれた機械語実行データ部分 (RW segment) のサイズ。	
ZIセグメントサイズ (ZI segment Size)	4	実行領域のコードが実行されるときに確保すべきメモリのサイズ。	
リソースセグメントサイズ (Resource segment Size)	4	ファイルに含まれたリソース領域のサイズ。	
ライブラリ初期化関数アドレス (Lib init function address)	4	実行領域に含まれて実行直前に呼び出されるべき実行関数の位置。	
エントリーファンクション (Entry function)	4	本ファイルに含まれたプログラムの実行開始位置。	
有効期限 (Expire Date)	2	本ファイルの実行有効期間。	30
使用回数 (Usage Count)	4	本ファイルの実行可能回数。	
DL	1	本ファイルの実行時に使用する動的リンクライブラリ (DLL) の個数。	
EH	1	静的フックイベント (Static Hooking Event) の個数。	
FA	1	ヘッダの可変領域に含まれた file association string の長さ。	
MM	1	ヘッダの可変領域に含まれた MIME string の長さ。	
UR	1	ヘッダの可変領域に含まれた file update URL sting の長さ ファイルの更新が必要な場合、可変領域の update URL を使用する。	
MS	1	ヘッダの可変領域に含まれた Manufacturer string の長さ；すなわち、本ファイルの製造者を表す文字列の長さ。	40
CS	1	ヘッダの可変領域に含まれた Copyright string の長さ。	

【 0 0 1 7 】

一方、ヘッダの可変フィールドは下記表 2 のようなデータ (フィールド) を含んでいる。

【 0 0 1 8 】

【 表 2 】

フィールド名	大きさ (byte)	特性または機能
インポートライブラリID (Import Library ID)	4	プログラムが使用する動的ライブラリのID。
ライブラリリビジョン (Library revision)	2	指定された動的ライブラリの最低有効バージョン。指定されたバージョン以上のバージョンの動的ライブラリでなければ、アプリケーションによって使用できない。
フックイベントクラス (Hook Event Class)	2	アプリケーションがフックするイベントのクラス。
フックイベントID (Hook Event ID)	2	アプリケーションがフックするイベントのID。
ファイル関連ストリング (File Association string)	可変	区分文字で区別される 拡張子を有する文字列。
MIMEストリング (MIME string)	可変	区分文字で区別される MIME ストリングを有する文字列。
更新URLストリング (Update URL string)	可変	ファイルの更新が必要な場合、参照するサイトのURL。
製造者ストリング (Manufacturer string)	可変	ファイルの製造者を表示する文字列。
著作権(者)ストリング (Copyright string)	可変	ファイルに含まれた内容の著作権(者)を表示する文字列。

10

20

## 【0019】

チェックサムは、実行領域、リソース領域等の全データを例えば4バイト単位で加算して、アプリケーションIDの語長と同一長のみの長さで切り出した値である。例えば、下位4ビットのみを切り出すようにしてもよい。プログラムのコンパイルの時にチェックサムを求め、このヘッダのチェックサムフィールドに格納する。移動通信端末側でファイルの容量を4バイト単位で算出し、一定の手法で切り出した実際にダウンロードされたファイルのチェックサムと、このヘッダに予め格納されるチェックサムとを比較することで、第1に、そのファイルの全体が正常にダウンロードされ、格納されたか否かを確認することができる。第2に、実行直前までにこのファイルの一部にプログラムやデータ等がウイルス等悪意のあるプログラムによって追加されたり、また、削除されたりというファイルへの変更が行われていないことを確認できる。このように、本実施形態では、チェックサムを用いてファイルの信頼性を検証する。

30

## 【0020】

アプリケーションIDは、プログラマや、ダウンロードのためのサーバーや、通信回線のキャリアなどによって利用されるこのファイル(アプリケーション)を他のアプリケーションから識別するためのIDである。移動通信端末にとっては、対外的に利用すべきアプリケーションの名称となる。

## 【0021】

マジックナンバーは、移動通信端末のプラットフォームに取ってのファイルの名称であり、移動通信端末にとってのアプリケーションの番号である。本実施形態では、マジックナンバーは、人間が直接に値を求めることが複雑な番号としている。すなわち、本実施形態では、アプリケーションIDと、チェックサムとを二進数で論理演算し、その演算結果をマジックナンバーとする。プラットフォームがダウンロードしたアプリケーションを識別するには、このマジックナンバーを利用するため、移動通信端末を悪意のあるプログラムから保護することができる。すなわち、移動通信端末によってプログラムが起動されるためには、このマジックナンバーを知らなければならず、一方、チェックサムとアプリケーションIDとの論理演算結果をマジックナンバーとする場合、マジックナンバーを通常のコンピュータ操作で知ることは困難である。従って、メーカーやキャリアによって許容された手法以外の手法でダウンロード用のプログラムを作成することは困難である。本実施形態では、マジックナンバーのデータ構造をこのように工夫することで、移動通信端末を

40

50

悪意のあるプログラマから保護することができる。

【0022】

論理演算は、XORとしているが、XORの場合にはオーバーフローがない点等で利用しやすいが、論理和、論理積などでも良い。そして、論理演算を行う場合、2つの引数の桁数が同一であると全桁について演算がなされる。表1に示す例では、アプリケーションIDとチェックサムとは同一に4バイトとしている。チェックサムを4バイトとするには、4バイトで表現できる容量よりもファイル容量が大きい場合には、切り出しを行う。仮に小さい場合には、予め特定されたビット列を合成するなどして、4バイトに合わせるとよい。この切り出しや合成、追加をここでは編集という。

【0023】

このチェックサムの編集の仕組みを、秘密にしておくことで、マジックナンバーの悪意のある特定作業はより困難となる。また、アプリケーションIDの範囲やファイル容量の範囲毎にこの編集の仕組みを変化させることで、マジックナンバーの特定は困難となる。このように、一定の論理関係に基づいて編集の仕組みを変化させる場合を含む論理演算による編集を、「所定の編集」という。所定の編集では、アプリケーションIDとチェックサムとのXORと、編集の仕組みを変化させた論理演算とを含む。また、所定の編集は、チェックサムの切り出しや合成のみならず、他の論理演算を行うようにしても良い。

【0024】

また、マジックナンバーは、合計で8バイトとして、アプリケーションIDと上記所定の編集によるマジックナンバーとの合成としてもよい。これにより、アプリケーションIDをユニークとしておくと、マジックナンバーは必ずユニークとなる。

【0025】

有効期限(使用期間制限)は、このプログラムのコンパイル時か、またはダウンロード時にヘッダの有効期限フィールドに格納される。移動通信端末では、端末の内部時計等とこの有効期限とを比較することで、プログラムファイルの使用制限を制御する。

【0026】

使用回数は、有効期限と同様に、このプログラムのコンパイル時か、またはダウンロード時にヘッダの使用回数フィールドに格納される。移動通信端末では、プログラムの実行回数をメモリーに格納するか、又は無線通信回線を介してサーバー等に格納し、プログラムファイルの起動時にこの使用回数と比較することで、プログラムファイルの使用制限を制御する。

【0027】

実行領域は実行コードセグメント(R0 segment)と実行データセグメント(RW segment)に二分される。

実行コードセグメント(R0 segment)は、端末のプロセッサによって直接実行が可能な機械語コードを含む領域である。一般に、機械語コードはプログラムのソースコード(source code)を入力とするコンパイラ(compiler)を用いて生成される。このように生成された機械語コードをファイルの実行コードセグメント(R0 segment)に記憶していて、実行の際に端末のメモリーにロードし、或いはプロセッサによって直接アクセス可能な場合にはメモリーにロードする手続無しで直接実行される。

【0028】

実行データセグメント(RW segment)は、端末のプロセッサによって直接解釈可能な機械語形態のデータを有する領域である。実行コードセグメント(R0 segment)と同様の手続を経て生成され、実行時に必ず端末のメモリーにロードされなければならない。

【0029】

リソース領域は、実行領域に含まれた機械語プログラム(R0及びRW segment)が使用するデータを有する領域である。リソース領域に含まれたデータは文字列とアイコン及び画像(イメージ)データである。各データは言語別に区分されてリソース領域に含まれるが、このように言語別に区分されたデータの各集合をリソースセクション(resource section)という。リソース領域はこのように言語別に区分された1つ以上のリソースセクションとリ

10

20

30

40

50

ソースヘッダからなる。

【0030】

リソースセクションはプログラムが使用する言語に応じて選択的にメモリにロードされて使用される。従って、各使用言語別リソースセクションの該当言語コードを区分する情報がリソースヘッダに含まれており、この際、使用される言語コードはISO 637:1 988規格に従うとよい。

【0031】

図4は前記リソース領域のリソースヘッダ部分とリソースセクション部分の詳細構成図である。ここでは、機械語コードで利用しやすいHEXフォーマットとしている。HEXフォーマットは、16ビット又は32ビット等で、例えば1バイトを上下の4ビットに分けて16進数2桁で表現する。16進数は2進数への変換が容易で、プロセッサの処理時間が短い。

10

本実施形態における「リソース領域」は、アプリケーションの実行コードが参照するデータを格納する領域であって、ダウンロードしたアプリケーションの実行時に実行コードがリソース領域を参照するためにトークンIDを利用する。すなわち、1つのトークンIDは1つのリソースデータを意味し、実行コードはリソース領域(リソースセクション)から1つのトークンIDに該当するリソースを抽出して使用する。通常、多国語の支援のために、リソース領域では使用言語別にリソースセクションが区分されており、1つのトークンIDに対しても使用言語数だけのリソースが存在する。すなわち、1つの言語を支援するリソースセクション#0に全てn個のリソースがある。従って、実行コードは支援する言語

20

【0032】

図5は本発明に係るファイルダウンロード及び実行過程を示すフローチャートである。まず、所定のプラットフォームを備えた移動通信端末を利用し、移動通信網及びインターネット網を介して前記構造の特定のファイルをダウンロードする(S511, 第1工程)。このようなファイルのダウンロードはインターネット網と移動通信網及び所定のゲートウェイ(WAPゲートウェイ)を用いる無線インターネットサービスで一般的に行われるものなので、その詳細な説明は省略する。

【0033】

端末のプラットフォームはダウンロードしたファイルのヘッダに含まれた所定のデータを用いてファイルの信頼性又は有効性を検証する(S512, 第2工程)。ファイルの信頼性検証には、チェックサム(checksum)を用いる。プラットフォームは、ダウンロードされたファイルの容量を確認し、その容量を示す数値の例えば下位4バイトを切り出す。この切り出された数値と、ファイルのヘッダに格納されたチェックサムとを比較し、一致する場合には信頼性有りとする。本実施形態では、チェックサムを固定長としているため、ファイルの容量を示す数値から切り出しを行っているが、ヘッダのチェックサムフィールドを可変長とする場合には、全体を比較するようにしても良い。

30

ここでは、ファイルの信頼性検証は、ファイルヘッダサイズ(File header size)フィールドからファイルの残り部分まで全体内容を4バイト単位で加算し、その結果の下位4バイトのみを取った値がヘッダに記憶されたチェックサム4バイトと同一の場合にのみプログラムの信頼性があると判定する。

40

【0034】

ファイルの有効性検証には、前述したヘッダのマジックナンバー(Magic number)、チェックサム(checksum)及びアプリケーションID(Application ID)を使用する。上述したように、プラットフォームは、チェックサムとアプリケーションIDとに所定の編集を行うことで、論理演算値を算出する。そして、その論理演算結果とマジックナンバーとを比較することで、そのファイルの有効性検証を行う。また、プラットフォームは、予め定められたプログラムのみを実行する場合には、実行可能なファイルのマジックナンバーを書き換え不能なROM等に格納しておき、マジックナンバーが一致する場合であっても、ROMに格納されたマジックナンバーと異なる場合には実行を許容しないようにしても良い。こ

50

のROMに格納する例では、当該移動通信端末の販売後に完成するプログラムファイルの実行が不可能となってしまう。従って、プログラムファイルのデータ構造として、マジックナンバーの範囲が一定範囲となるようにチェックサムとの関係でアプリケーションIDを特定するようにし、プラットフォームはマジックナンバーが一定範囲でない場合（例えば、マジックナンバーが正のとき）には実行を許容しないようにしても良い。

ここでは、チェックサムとアプリケーションIDの各4バイトをXOR（exclusive OR，排他的論理和）し、ヘッダに記録されているマジックナンバーと比較して同一の場合に限ってファイルの有効性があると判定する。

【0035】

このようなファイルの有効性及び信頼性検証が両方とも認められた場合に限って、ファイルの実行を許容する（S513）。また、1つの検証のみを行うか、通過しても特別な場合には実行を許容するようにすることもできる。

【0036】

このように、ファイルの有効性及び/又は信頼性が確認された場合にファイルを実行する。ファイルの実行は、基本的な通信機能等によって使用される必要なメモリを別に確保した状態で、ファイルの実行領域（RO及びRWセグメント等）に格納されたコードをメモリにロードし、或いは直接アクセスが可能な場合にはメモリのローディング過程なしで実行ビュー（Execution View）を作る（S516，第3工程）。実行ビューとは、プロセッサによってプログラムが実行できる形態でメモリにロードされるか、或いはメモリロード無しで実行準備が出来た状態を意味する。

【0037】

すなわち、このステップS516では、実行ビュー生成工程として、第1に、ファイルヘッダのインポートライブラリID等に基づいて予めプラットフォームに格納され又は同時にダウンロードしたライブラリとの関係でリンクやアドレスの特定を行うことで、ライブラリを使用可能にする。第2に、RO及びRWセグメントのアドレス方式が実アドレスではない場合には、機械語コードの修飾アドレス等を当該プラットフォームのメモリのアドレスに割り当てる。このようなライブラリやプラットフォームのメモリアドレスとの関係を整理し、機械語コードを実行可能な状態とする。また、第3に、ファイルヘッダのZIセグメントサイズが格納されている場合には、当該機械語コードによって使用されるメモリをプラットフォームのメモリから確保する。ZIセグメントサイズが格納されていない場合には、可能なメモリを当該機械語コードに割り当てる。使用メモリを確保できない場合には、プログラムの実行は許容されない。

【0038】

機械語プログラムをダウンロードする移動通信端末で当該プログラムを実行する場合には、ステップS516が、プログラムファイルのプラットフォームへの格納形態に応じてメモリへのロード又は端末プロセッサの直接アクセスを制御すると共に、当該機械語コードに前記プラットフォームのメモリアドレスを割り当て及び/又は当該プラットフォームのライブラリを使用可能に設定する実行ビュー生成工程を備えると良い。

【0039】

また、端末のプラットフォームが実行ビューを生成する際に、端末に設定された使用言語の設定に応じてリソース領域の当該リソースセクションをメモリにロードする（S517）。

【0040】

生成された実行ビューと、ロードされたリソースセクションを用いてファイルの所定の機能を具現することにより、ファイルを実行する（S518）。ファイル実行に成功すると、プラットフォームは、使用回数のモニタリングとして、ファイル実行回数に1を加算して、メモリ3等に実際の使用回数を格納する。

【0041】

前記ファイルヘッダの有効期限（Expire date）または使用回数（Usage Count）に所定の値が割り当てられることにより、使用可能期間又は使用可能回数が制限されている場合に

10

20

30

40

50

は ( S 5 1 4 )、プラットフォームは前記ファイルの使用回数又は使用期間をモニタリングし、ファイルヘッダに含まれた使用可能期間又は使用可能回数を超えれば実行を許容しないようにする工程をさらに含むことができる ( S 5 1 5、S 5 2 2、第 4 工程 )。

【 0 0 4 2 】

また、ファイルヘッダの UR 及び Update URL string フィールドに値が与えられて自動更新機能 ( Updating ) がある場合 ( S 5 1 9 )、更新条件になったか或いはユーザからの更新要請があれば ( S 5 2 0 )、ファイルヘッダにある更新 URL スtring ( Update URL string ) を用いて該当ファイルをダウンロードするサイトに接続し、そのサイトから更新のためのデータをダウンロードして更新 ( アップデート ) する工程 ( S 5 2 1、第 5 工程 ) をさらに含むことができる。

10

更新条件の例としては、所定の期間が経過した後アップデートする場合や、所定の回数だけ実行した後アップデートする場合などが挙げられる。

【 0 0 4 3 】

図 6 は、本実施形態による特定のデータ構造のプログラムファイルをダウンロードして実行する移動通信端末の機能構成を示すブロック図である。本実施形態による移動通信端末は、プロセッサ又は論理回路の制御に従ってファイルのダウンロード等の通信を制御する無線通信部 1 と、ディスプレイ及びキーパッド等のユーザインタフェース 2 と、プログラムファイル及びデータを記憶するメモリ 3 と、このメモリ 3 を使用してプログラムファイルを実行するプロセッサ 4 と、実行部 ( プロセッサ ) 4 と、ダウンロードプログラムファイルを管理するためのダウンロードプログラムファイル制御部 1 0 とを備えている。

20

【 0 0 4 4 】

そして、ダウンロードプログラムファイル制御部 1 0 は、ダウンロード制御部 1 1 と、更新制御部 1 2 と、実行制御部 1 2 とを備えている。ダウンロード制御部 1 1 は、プロセッサで直接実行できる機械語コードと、少なくとも当該機械語コードのマジックコード、チェックサム及びアプリケーション ID とを格納するヘッダとを有する特定のデータ構造のプログラムファイルをダウンロードすると共に、メモリ 3 に格納する。

【 0 0 4 5 】

実行制御部 1 2 は、本実施形態では特に、ステップ S 5 1 2 及び 5 1 3 のうち有効性に関する処理を実行する有効性検証機能 1 4 と、当該工程のうち信頼性に関する処理を実行する信頼性検証機能 1 5 と、ステップ S 5 1 4、5 1 5 の処理を実行する使用制限機能 1 6 と、ステップ S 5 1 6 の実行ビュー生成に関する処理を実行する実行ビュー生成部 1 7 と、ステップ S 5 1 7 の処理を実行するリソースデータ読込制御部 1 8 とを備えている。

30

【 0 0 4 6 】

有効性検証機能 1 4 は、実行指令が入力された時に、メモリ 3 に格納されたプログラムファイルを当該プログラムファイルのヘッダに含まれた各データ ( チェックサム、アプリケーション ID、マジックナンバー等 ) を用いて当該プログラムファイルの有効性を検証する。例えば、ファイルの容量を示す数値のうちヘッダのアプリケーション ID と同一長さの数値を切り出し、その数値とアプリケーション ID との XOR を求める。この XOR の演算結果と、ヘッダに予め格納されたマジックナンバーとが一致する場合に、有効性有り と判定し、不一致の場合には信頼性なしと判定する。有効性検証機能 1 4 は、上記所定の編集を行う編集機能をさらに備えるようにしても良い。

40

【 0 0 4 7 】

信頼性検証機能 1 5 は、実行指令が入力されたときに、まず、メモリ 3 に格納されたプログラムファイルの容量を測定すると共に、当該容量を示す数値のうちヘッダのチェックサムと同一長さの数値から、予め定められた部分 ( 例えば、下位 4 バイト ) を切り出す。続いて、切り出した数値と、ヘッダに予め格納されたチェックサムとを比較する。チェックサムが一致する場合には、信頼性有り と判定し、不一致の場合には信頼性なしと判定する。

【 0 0 4 8 】

使用制限機能 1 6 は、使用期間制限に関しては、現在日時を内部時計等から読み出し、へ

50

ッダに含まれる有効期限 (Expire Date) と比較して、有効期限を過ぎている場合には、実行を不許可とする。使用回数に関しては、メモリ 3 等に格納した実際の使用回数と、ファイルヘッダに予め格納された使用回数 (Usage Count) とを比較して、実際の使用回数がヘッダに定義された使用回数と同一又は越える場合に、実行を不許可とする。

【 0 0 4 9 】

実行ビュー生成部 1 7 は、Z I セグメントに格納されたメモリサイズのメモリ空間をメモリ 3 への確保を試み、確保できない場合には実行を不可能とする。また、R O セグメントに格納された実行コードのアドレス指定方式によっては、当該実行コードに実アドレスを割り当てる。さらに、ヘッダの D L 等を参照して、ライブラリを使用可能にするための処理を行う。そして、ダウンロードされたプログラムファイルのうち、R O 及び R W セグメントの実際の格納位置が、プロセッサ 4 によって直接に実行可能なアドレスであれば、ライブラリ初期化関数アドレス等との関係でプロセッサに機械語コードの先頭アドレスを渡す。一方、直接に実行できないアドレスに格納されている場合には、実行可能なメモリへのロードを行い、そのアドレスをプロセッサに渡す。

10

【 0 0 5 0 】

リソースデータ読込制御部 1 8 は、リソースデータの国コード等に基づいて、使用可能言語を判定し、使用可能言語に対応するリソースデータの読込を制御する。

【 0 0 5 1 】

更新制御部 1 2 は、ステップ S 5 1 9 , 5 2 0 及び 5 2 1 の処理を行う。この自動更新のタイミングは、例えば、所定の期間 (有効期限) が経過した後アップデートする場合や、所定の回数 (使用回数) だけ実行した後アップデートする場合などがある。これにより、期間や回数での課金をスムーズに行うことができる。

20

【 0 0 5 2 】

実行部 (プロセッサ) 4 は、有効性等が検証された場合に限り、当該プログラムファイルの機械語コードをメモリ 3 にロードし或いは当該プロセッサ 4 が直接にアクセスして、当該プログラムファイルの機械語コードを実行する。

実行部 4 は、移動通信端末に装備された C P U 及び論理回路であり、どの機能の部分を論理回路で実装するかは任意である。

【 0 0 5 3 】

図 6 に示す構成や、図 5 に示す処理は、移動通信端末用のプロセッサが、ダウンロードファイルの管理及び実行を制御するためにダウンロードプログラムファイル管理用プログラムを実行することで実現できる。このダウンロードプログラムファイル管理用プログラムは、例えば移動通信端末の R O M に格納しておくようにしてもよい。

30

【 0 0 5 4 】

このダウンロードプログラムファイル管理用プログラムは、前記移動通信端末を動作させる指令として、下記の指令を備える。この指令は、図 5 に示す各工程や、図 6 に示す各制御部及び各機能と対応している。

プロセッサで直接実行できる機械語コードと、少なくとも当該機械語コードのマジックコード、チェックサム及びアプリケーション I D とを格納するヘッダとを有する特定のデータ構造のプログラムファイルをダウンロードすると共に、前記メモリに格納する第 1 指令。この第 1 指令は、図 5 のステップ S 5 1 2 等に対応し、図 6 のダウンロード制御部 1 1 に対応する。

40

プログラムファイルのヘッダに含まれる所定のデータを用いて、当該ファイルの信頼性及び/又は有効性を検証させる第 2 指令。この第 2 指令は、図 5 のステップ S 5 1 2 , 5 1 3 に対応し、図 6 の有効性検証機能 1 4 及び信頼性検証機能 1 5 に対応する。第 2 指令は、実際のファイル容量とアプリケーション I D とを使用した所定の編集をプロセッサにさせる指令を備えるようにしても良い。

前記第 2 指令に応じて有効性及び/又は信頼性が検証される場合に限り、当該プログラムファイルの機械語コードをメモリ 3 にロードし或いはプロセッサ 4 が直接にアクセスして、当該プログラムファイルの機械語コードを実行させる第 3 指令。この第 3 指令は、図

50

5のステップS516, 517及び518に対応し、図6の実行ビュー生成部17, リソースデータ読込制御部18及び実行部4に対応する。従って、第3指令は、より詳細な指令として、プロセッサ4に実行ビューを生成させる実行ビュー生成指令を備えるようにしても良い。

【0055】

【発明の効果】

以上述べたように、本発明に係るファイル構造方法及び端末によれば、ファイルの実行領域を、端末によって直接実行可能な機械語コードで構成することにより、次のような利点を有する。

機械語コードは、バイトコード基盤のプログラム(ファイル)に比べてサイズが小さく、ダウンロードに必要な時間が短縮され、ダウンロードしたファイルを格納するためのメモリ空間も小さくなる。

また、機械語コードは、プロセッサによって直接実行が可能なので、バイトコード基盤のプログラムに比べて実行速度が速い。通常、機械語プログラムコードはコンパイルまたは変換(インタラプト)の必要なプログラムより約8~20倍程度が速い。

また、リソース領域をプログラムの実行領域と分離し、容易な多国語支援のために言語別データの集合を区分して構成することにより、実行時に必要なデータのみをメモリにロードして使用することができ、よって、実行時にかかるメモリ要求量を減少させることができる。

ダウンロードに必要な時間が短く、実行が高速で、メモリ要求量が少ないため容量の大きいメモリを移動通信端末に装備する必要がないことから、本発明では、バッテリーの連続稼働時間を短縮させずに、プログラムファイルのダウンロード機能を移動通信端末に付加することができる。特に、バイトコード基盤の仮想端末となる移動通信端末と比較して、実行速度及びメモリ空間との関係から、バッテリーの必要な充電回数を少なくし、連続して稼働できる時間を相対的に長くすることができる。

ファイルのヘッダを用いることにより、ファイルの有効性の検証を行うことができ、使用を制限し或いは自動アップデートを可能にする。特に、本発明では、マジックコードが、前記実行部の論理演算命令を使用した場合と同一の論理演算結果となる論理演算を用いて、前記アプリケーションIDとチェックサムとに論理演算をした論理演算値であって、有効性検証機能が、前記実行指令が入力された時に、当該プログラムファイルのヘッダの前記アプリケーションIDと前記チェックサムとに前記実行部の論理演算をし、前記プログラムファイルのヘッダのマジックコードと当該論理演算結果とを比較して、一致した際に有効性有りとするため、ヘッダの情報と、移動通信端末が実行できる論理演算とにより、メーカーやキャリアがアプリケーションIDやマジックナンバーのデータ構造を工夫することで、悪意のあるプログラマから移動通信端末を保護しつつ、事後的な機械語コードのプログラムの更新や導入をすることができる。

このように、本発明によると、ファイルのヘッダの各フィールドを用いることで多様な機能を実現しつつ、機械語コード実行による利点を、移動通信端末で実現することができる。

【図面の簡単な説明】

【図1】従来の移動通信端末用ファイルの内部構成を示す図である。

【図2】本発明に使用される移動通信端末用ファイルの内部構成を示す図である。

【図3】本発明に使用される端末用ファイルのヘッダ部分に対する詳細図である。

【図4】本発明に使用される端末用ファイルのリソース領域に対する詳細図である。

【図5】本発明に係る端末用ファイルのダウンロード及び実行過程を示すフローチャートである。

【図6】本発明に係る端末用ファイルを実行する移動通信端末の機能構成を示すブロック図である。

【符号の説明】

1 無線通信部

10

20

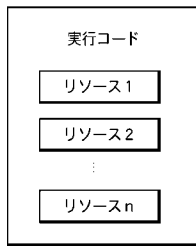
30

40

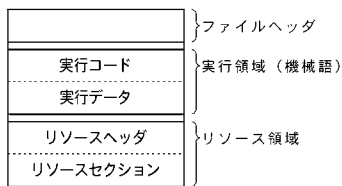
50

- 2 表示部
- 3 メモリ
- 4 プロセッサ (実行部)
- 10 ダウンロードプログラムファイル制御部
- 11 ダウンロード制御部
- 12 更新制御部
- 13 実行制御部
- 14 有効性検証機能
- 15 信頼性検証機能
- 16 使用制限機能
- 17 実行ビュー生成部
- 18 リソースデータ読込制御部

【図1】



【図2】



【図3】

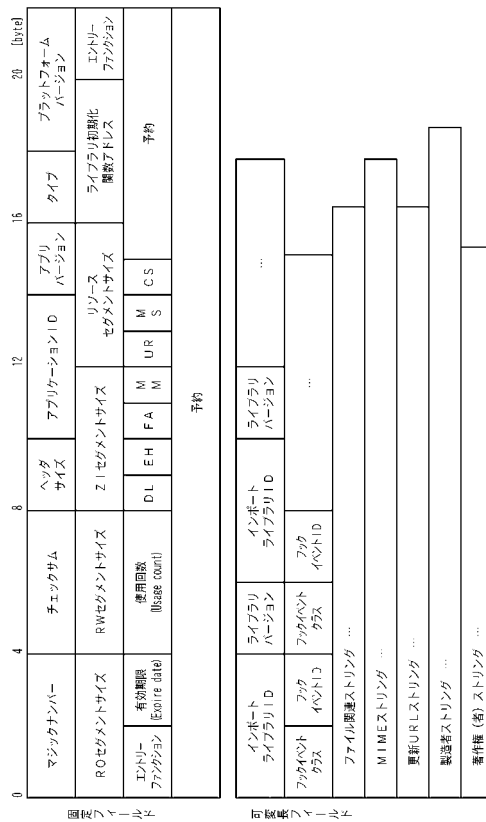


図3のフィールド

図3のフィールド



---

フロントページの続き

(72)発明者 金 容勳

大韓民国京畿道安養市東安區達安洞 1101-2 サエトビュルアパート102-305

審査官 石川 正二

(56)参考文献 特開2001-256057(JP,A)

特開2001-056784(JP,A)

特開2001-249815(JP,A)

国際公開第01/059563(WO,A1)

安井晴海, 携帯JavaとBREW, 日経コミュニケーション, 日経BP社, 2001年 6月  
18日, 第344号, p.140~147

(58)調査した分野(Int.Cl., DB名)

G06F 9/445

G06F 11/10

G06F 21/22