

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第6318113号  
(P6318113)

(45) 発行日 平成30年4月25日(2018.4.25)

(24) 登録日 平成30年4月6日(2018.4.6)

(51) Int.Cl.

F I

G O 6 F 13/00 (2006.01)

G O 6 F 13/00 5 4 0 C

G O 6 F 12/00 (2006.01)

G O 6 F 13/00 5 4 0 B

G O 6 F 12/00 5 4 6 M

請求項の数 4 (全 14 頁)

(21) 出願番号 特願2015-87362 (P2015-87362)  
 (22) 出願日 平成27年4月22日(2015.4.22)  
 (62) 分割の表示 特願2013-150483 (P2013-150483)  
                   の分割  
           原出願日 平成25年7月19日(2013.7.19)  
 (65) 公開番号 特開2015-165416 (P2015-165416A)  
 (43) 公開日 平成27年9月17日(2015.9.17)  
           審査請求日 平成28年7月15日(2016.7.15)

(73) 特許権者 599115217  
                   株式会社 ディー・エヌ・エー  
                   東京都渋谷区渋谷二丁目21番1号  
 (74) 代理人 110001210  
                   特許業務法人 Y K I 国際特許事務所  
 (72) 発明者 多久島 信隆  
                   東京都渋谷区渋谷二丁目21番1号 株式  
                   会社 ディー・エヌ・エー内

審査官 小林 義晴

最終頁に続く

(54) 【発明の名称】 情報端末及びデータ処理プログラム

(57) 【特許請求の範囲】

【請求項 1】

内蔵された処理部が実行するネイティブ部分実行部と、  
 ネットワークを介して接続されたサーバから受けた情報の表示処理を行うウェブブラウ  
 ザと、

が組み合わされたアプリケーションを実行する情報端末であって、

前記ネイティブ部分実行部及び前記ウェブブラウザを実行可能な実行手段は、  
 前記ネイティブ部分実行部にてネットワークを介してサーバにアクセスして通信を行い  
 、前記サーバから情報を取得可能な通信手段と、  
 前記通信手段によって取得した情報を記憶可能な記憶手段と、  
 を備え、

前記記憶手段は、前記ウェブブラウザからアクセス可能な第1記憶手段と、前記ネイテ  
 イブ部分実行部からアクセス可能な第2記憶手段と、を含み、

前記第1記憶手段及び前記第2記憶手段の少なくとも一つに前記表示処理に必要なアセ  
 ットデータが記憶されている場合には、記憶されている前記アセットデータを前記ウェブ  
 ブラウザに渡し、

前記第1記憶手段に前記アセットデータが記憶されていない場合、前記通信手段によっ  
 て、前記サーバから前記アセットデータのバージョン情報を取得し、前記バージョン情報  
 を用いて前記第2記憶手段に前記アセットデータが記憶されていないと判別された場合に  
 、前記サーバにアクセスして前記アセットデータを取得し、取得した前記アセットデータ

を前記ウェブブラウザに渡すことを特徴とする情報端末。

【請求項 2】

請求項 1 に記載の情報端末であって、

前記サーバから取得された前記アセットデータを前記第 2 記憶手段に記憶させることを特徴とする情報端末。

【請求項 3】

請求項 1 又は 2 に記載の情報端末であって、

前記通信手段は、前記サーバから取得される前記アセットデータを前記ウェブブラウザで処理可能なデータ形式に変換することを特徴とする情報端末。

【請求項 4】

コンピュータ端末内の処理部が実行するネイティブ部分実行部と、

前記コンピュータ端末とネットワークを介して接続されたサーバから受けた情報の表示処理を行うウェブブラウザと、

が組み合わされたアプリケーションを含むデータ処理プログラムであって、

前記コンピュータ端末を、

前記ネイティブ部分実行部及び前記ウェブブラウザを実行可能な実行手段は、

前記ネイティブ部分実行部にてネットワークを介してサーバにアクセスして通信を行い、

前記サーバから情報を取得可能な通信手段と、

前記通信手段によって取得した情報を記憶可能な記憶手段と、

を備え、

前記記憶手段は、前記ウェブブラウザからアクセス可能な第 1 記憶手段と、前記ネイティブ部分実行部からアクセス可能な第 2 記憶手段と、を含み、

前記第 1 記憶手段及び前記第 2 記憶手段の少なくとも一つに前記表示処理に必要なアセットデータが記憶されている場合には、記憶されている前記アセットデータを前記ウェブブラウザに渡し、

前記第 1 記憶手段に前記アセットデータが記憶されていない場合、前記通信手段によって、前記サーバから前記アセットデータのバージョン情報を取得し、前記バージョン情報を用いて前記第 2 記憶手段に前記アセットデータが記憶されていないと判別された場合に、前記サーバにアクセスして前記アセットデータを取得し、取得した前記アセットデータを前記ウェブブラウザに渡す情報端末として機能させることを特徴とするデータ処理プログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、サーバから情報を取得する情報端末及びその処理を実現するデータ処理プログラムに関する。

【背景技術】

【0002】

近年、スマートフォン等の携帯端末において、端末内の処理部が実行するネイティブ部分実行部と、ネットワークを介して接続されたサーバ側（ウェブ部分実行部）で処理された情報を受信して利用するウェブブラウザと、を組み合わせたアプリケーションを用いた処理が利用されている。

【0003】

このようなアプリケーションを用いることによって、携帯端末に固有に搭載されている機能をネイティブ部分実行部によって十分に活用できると共に、頻繁に更新される処理はウェブ部分実行部に負担させる等、ネイティブ部分実行部とウェブ部分実行部の両方の利点を得ることができる。

【0004】

図 6 は、ネイティブ部分実行部とウェブ部分実行部との処理で実現されるアプリケーシ

10

20

30

40

50

ョンを用いた携帯端末100とサーバ102との情報のやり取りを説明する図である。携帯端末100は、アプリケーションの実行によって実現されるウェブブラウザ10に含まれるダウンロード10aを用いて、ネットワーク104を介して接続されたサーバ102にアクセスする。ダウンロード10aは、例えば、Base64等でエンコードされた情報をサーバ102から受信する。サーバ102から受信された情報は、携帯端末100のウェブブラウザ10からアクセス可能なブラウザキャッシュ10b、又は、ネイティブ12側のストレージ12aのいずれかに格納及び保持される。このとき、ネイティブ部分実行部の実行によりウェブブラウザ10から情報を受けてストレージ12aにその情報を格納し、ウェブブラウザ10から情報の要求を受けてストレージ12aに保持されている情報をウェブブラウザ10へ受け渡す処理が実現される。ウェブブラウザ10とストレージ12aとの間においてもBase64等でエンコードされた情報が受け渡しされ、ストレージ12aにはエンコードされた状態で情報が格納される。

10

#### 【0005】

また、ウェブブラウザ10のキャッシュ機能を用いて、サーバ102から一度取得したデータをブラウザキャッシュ10bに記憶しておき、ブラウザキャッシュ10bに記憶されているデータについて再度取得する手間を省く処理が知られている。

#### 【0006】

非特許文献1には、ブラウザキャッシュに記憶するデータの最終更新日とバージョン情報とを関連付けしたハッシュ値等を管理しておくことで、取得しようとするデータがキャッシュに記憶されている場合、通信ネットワークを介してそのデータを取得することなく、キャッシュに記憶されたデータを利用する技術が開示されている。

20

#### 【0007】

例えば、ネットワークを介してサーバと接続されている情報端末を利用したゲームにおいて、頻繁に更新されるキャラクタデータを情報端末へ送信する際のサーバと情報端末間での通信負荷を低減するために用いられている。

#### 【先行技術文献】

#### 【非特許文献】

#### 【0008】

【非特許文献1】<https://developers.google.com/speed/docs/best-practices/caching>

#### 【発明の概要】

30

#### 【発明が解決しようとする課題】

#### 【0009】

ところで、ネットワークの通信帯域の容量が小さい場合、ネットワークを介して画像データやトゥルータイプフォント(ttf)等の容量の大きなアセットデータを送受信する際にタイムアウトを起こしてしまうおそれがある。また、上記のように、ブラウザキャッシュやネイティブ側のストレージを利用したとしても、通信帯域が十分でなくなる場合がある。

#### 【0010】

このような状況において、情報端末とサーバとの間で情報を効率的に通信可能とする技術が望まれている。

40

#### 【課題を解決するための手段】

#### 【0011】

本発明の1つの態様は、内蔵された処理部が実行するネイティブ部分実行部と、ネットワークを介して接続されたサーバから受けた情報の表示処理を行うウェブブラウザと、が組み合わされたアプリケーションを実行する情報端末であって、前記ネイティブ部分実行部及び前記ウェブブラウザを実行可能な実行手段は、前記ネイティブ部分実行部にてネットワークを介してサーバにアクセスして通信を行い、前記サーバから情報を取得可能な通信手段と、前記通信手段によって取得した情報を記憶可能な記憶手段と、を備え、前記記憶手段に前記表示処理に必要なアセットデータが記憶されている場合には、記憶されている前記アセットデータを前記ウェブブラウザに渡し、前記記憶手段に前記アセットデータ

50

が記憶されていない場合には、前記通信手段によって前記サーバから前記アセットデータを取得し、取得した前記アセットデータを前記ウェブブラウザに渡すことを特徴とする情報端末である。

【0012】

ここで、前記記憶手段は、前記ウェブブラウザからアクセス可能な第1記憶手段と、前記ネイティブ部分実行部からアクセス可能な第2記憶手段と、を含み、前記第1記憶手段及び前記第2記憶手段の少なくとも一つに前記アセットデータが記憶されている場合には、記憶されている前記アセットデータを前記ウェブブラウザに渡し、前記第1記憶手段及び前記第2記憶手段に前記アセットデータが記憶されていない場合には、前記通信手段によって前記サーバから前記アセットデータを取得し、取得した前記アセットデータを前記ウェブブラウザに渡すことが好適である。

10

【0013】

また、前記第1記憶手段に前記アセットデータが記憶されていない場合、前記通信手段によって、前記サーバから前記アセットデータのバージョン情報を取得し、前記バージョン情報を用いて前記第2記憶手段に前記アセットデータが記憶されていないと判別された場合に、前記サーバにアクセスして前記アセットデータを取得することが好適である。

【0014】

また、前記サーバから取得された前記アセットデータを前記第2記憶手段に記憶させることが好適である。

【0015】

20

また、前記通信手段は、前記サーバから取得される前記アセットデータを前記ウェブブラウザで処理可能なデータ形式に変換することが好適である。

【0016】

上記情報端末は、以下のデータ処理プログラムにより実現することができる。すなわち、コンピュータ端末内の処理部が実行するネイティブ部分実行部と、前記コンピュータ端末とネットワークを介して接続されたサーバから受けた情報の表示処理を行うウェブブラウザと、が組み合わされたアプリケーションを含むデータ処理プログラムであって、前記コンピュータ端末を、前記ネイティブ部分実行部及び前記ウェブブラウザを実行可能な実行手段は、前記ネイティブ部分実行部にてネットワークを介してサーバにアクセスして通信を行い、前記サーバから情報を取得可能な通信手段と、前記通信手段によって取得した情報を記憶可能な記憶手段と、を備え、前記記憶手段に前記表示処理に必要なアセットデータが記憶されている場合には、記憶されている前記アセットデータを前記ウェブブラウザに渡し、前記記憶手段に前記アセットデータが記憶されていない場合には、前記通信手段によって前記サーバから前記アセットデータを取得し、取得した前記アセットデータを前記ウェブブラウザに渡す情報端末として機能させることを特徴とするデータ処理プログラムである。

30

【発明の効果】

【0018】

本発明によれば、アプリケーションにプロキシ機能を組み込むことによって、サーバと情報端末との間の通信量を低減し、効率良く情報を通信することが可能となる。

40

【図面の簡単な説明】

【0019】

【図1】本発明の実施の形態におけるデータ処理システムの構成を示す図である。

【図2】本発明の実施の形態におけるデータ処理システムの機能ブロックを示す図である。

【図3】本発明の実施の形態におけるデータの送受信を説明するフローチャートである。

【図4】ページの切り替えを説明する図である。

【図5】本発明の実施の形態におけるデータ処理システムの別例の機能ブロックを示す図である。

【図6】従来のデータ処理システムの機能ブロックを示す図である。

50

**【発明を実施するための形態】****【0020】**

以下、本発明の実施形態について、図面に基づいて説明する。

**【0021】****<システムの全体構成>**

図1は、本実施形態に係るデータ処理システムの概略構成を示すブロック図である。サーバ202は、ネットワーク204に接続されており、このネットワーク204に情報端末200が接続されている。情報端末200は、ネットワーク204を介し、必要なデータをサーバ202から入手してゲーム等の処理を実行する。

**【0022】**

なお、本実施の形態では、情報端末200においてゲームを実行する例について説明するが、本発明の適用範囲はこれに限定されるものでなく、サーバ202と情報端末200との間で随時更新されるデータの通信を行う態様であれば適用範囲となる。

**<サーバの構成及び機能>**

サーバ202は、コンピュータであり、各種のデータ処理機能、通信機能、記憶機能を備えている。サーバ202には、ゲームロジック、およびそのゲームを実行する際に使用されるユーザデータ、およびアセットデータが記憶される。

**【0023】**

アセットデータは、たとえば、画像データ、動画（アニメーションデータ）、文字情報（トゥールタイプフォント：t t f）などを含む。アセットデータは、これらのデータに限定されるものではないが、文書データ等よりもデータ容量が大きいデータである場合に本発明の効果がより顕著となる。

**【0024】**

一般的に、ソーシャルゲームでは、ゲームを実行するためのゲームロジックは基本的に変更が頻繁でないが、画像データ、動画（アニメーションデータ）などのアセットデータは随時変更される。すなわち、アセットデータは開発者等によって随時変更され、メモリ20内の既存のアセットデータが新規のアセットデータによって随時書き換えられ、更新される。このように変更を受けたアセットデータを情報端末200で実行中のゲームにも反映するために、サーバ202から情報端末200へアセットデータを送信する必要がある。

**【0025】**

なお、開発者のコンピュータは、ネットワーク204を介しサーバ202に接続してもよいが、内部LAN、無線通信など他の通信回線を介しサーバ202に接続してもよい。

**【0026】**

サーバ202は、図2の機能ブロック図に示すように、データ更新部22を有し、このデータ更新部22がメモリ20内のデータの更新を管理する。すなわち、データ更新部22は、更新データを受け取った場合に、メモリ20内のデータを更新する。ここで、このデータ更新は、基本的にアセットデータの更新を前提としている。ここで、アセットデータは、1つの画像、1つの動画等を1単位として、それぞれ1つのファイルとして記憶されており、各ファイルにはファイル名その他、更新時刻、ファイル容量などの属性情報が付与されている。すなわち、既存のアセットデータと新規のアセットデータとは、この属性情報に基づいて対応付けすることが可能である。たとえば、メモリ20に記憶されているアセットデータを更新する際には、古いアセットデータを新しいアセットデータで更新し、新しいアセットデータに更新された古いアセットデータと同一のファイル名を付与することにより対応付けが行われる。

**【0027】**

サーバ202は、データを送信する場合に、その送信データについて、ファイル毎にハッシュ値を計算し、計算結果のハッシュ値をファイルとともに送信する。

**【0028】**

データ更新部22は、ファイルのデータ自体や属性情報からハッシュ値を計算し、これ

10

20

30

40

50

をファイルと関連づけて送信する。なお、ハッシュ値は、そのファイルに含まれるアセットデータ（画像データ、動画データ等）又はそれに関する属性情報から算出されればよいが、アセットデータ自体から算出することが好適である。すなわち、アセットデータが更新されると、アセットデータ自体から算出されたハッシュ値は更新前のアセットデータ自体から算出されたハッシュ値とは異なる値となり、ハッシュ値はアセットデータ自体のバージョンを示すバージョン情報となる。このハッシュ値を比較することにより、アセットデータが更新されたか否かを確実に把握することができる。

#### 【 0 0 2 9 】

ここで、ファイルが更新される度に、当該ファイルのハッシュ値を計算して、ファイルと対応付けてメモリ 20 に記憶することも好適である。こうすることによって、ファイルの送信要求が生じた場合に、すでに記憶されているハッシュ値を送信することもできる。また、一定周期毎に、メモリ 20 内のファイルのハッシュ値を計算して更新してもよく、さらにファイルを送信したときに再計算したハッシュ値をファイルに対応付けて更新してもよい。

10

#### 【 0 0 3 0 】

なお、本実施の形態では、1つのファイルを単位データとして処理する例を示すが、複数ファイルを含む単位データとして当該単位データ毎にハッシュ値を算出して処理してもよい。

#### 【 0 0 3 1 】

また、本実施形態では、バージョン情報としてハッシュ値を適用したが、変更されたアセットデータ（ファイル）に応じて変更されるバージョン情報としてもよい。たとえば、バージョン情報として、チェックサム、チェックディジット、フィンガープリント等としてもよい。

20

#### 【 0 0 3 2 】

アセットデータの作成者は、例えば、自己のコンピュータをサーバ 202 に接続し、メモリ 20 内のデータを随時書き換える。データ更新部 22 は、情報端末 200 からデータの提供を要求された際に、送信対象となるファイルを含む単位データに基づきそのハッシュ値を計算して、情報端末 200 に送信する。

#### 【 0 0 3 3 】

たとえば、データ更新部 22 に下記のような A P I を用意しておく。

30

```
__get__bundle?paths=a,b,c,d - - -> {a: {hash
: md5, body: base64} . . . }
```

情報端末 200 からのアクセスによって、この A P I が呼ばれた時にハッシュ値 m d 5 を計算してファイル a のバージョン情報とする。このようにすることで、ファイル a が差し替えられた場合には、その後に該当ファイルをユーザに送信すると、その送信データに対応するハッシュ値 m d 5 値が情報端末 200 に送信され、情報端末 200 において記憶される。

#### 【 0 0 3 4 】

< 情報端末の構成及び機能 >

情報端末 200 は、携帯電話、スマートフォン、タブレット等のコンピュータを含む端末装置である。情報端末 200 は、図 1 に示すように、C P U を含む処理部 30 が基本的なデータ処理を行う。また、情報端末 200 には、入力部 32、表示部 34 が設けられ、必要なデータ入力、各種表示を行う。たとえば、表示部 34 は液晶や有機 E L パネルであり、入力部 32 は表示部 34 と共に設けられたタッチパネルとすることができる。また、情報端末 200 には、記憶部 36 が設けられる。記憶部 36 は、半導体メモリやフラッシュメモリ等を含み、情報端末 200 によって処理される情報（アセットデータを含む）を格納及び保持する。記憶部 36 は、後述するブラウザキャッシュやネイティブキャッシュとして利用される。

40

#### 【 0 0 3 5 】

情報端末 200 では、アプリケーションが実行され、ネットワーク 204 を介したサー

50

バ 2 0 2 へのアクセスや情報端末 2 0 0 での情報処理が行われる。例えば、情報端末 2 0 0 のユーザがゲームを開始する場合には、URL を特定してそのゲームプログラムの提供先であるサーバ 2 0 2 にアクセスし、ここからゲームプログラムを格納するアプリケーションをダウンロードする。本実施の形態では、アプリケーションによりゲームの処理が行われる。

#### 【 0 0 3 6 】

ここで、アプリケーションとは、情報端末 2 0 0 に内蔵された処理部 3 0 が実行して情報端末 2 0 0 において情報の処理を行わせるネイティブ部分実行部と、サーバ 2 0 2 からネットワーク 2 0 4 を介して受信された情報を処理するウェブブラウザと、を組み合わせ

10

#### 【 0 0 3 7 】

アプリケーションであるゲームプログラムは、記憶部 3 6 のネイティブキャッシュ（ネイティブストレージ）3 8 に記憶され、処理部 3 0 がゲームプログラムを実行することで、ウェブブラウザ 4 0、プロキシ 4 2 及びダウンロード 4 4 の機能が実現される。情報端末 2 0 0 は、これらの機能によってサーバ 2 0 2 と通信し、必要な画像、動画などのアセットデータを取得し、表示部 3 4 に表示しつつゲームを進行させる。

#### 【 0 0 3 8 】

プロキシ 4 2 及びダウンロード 4 4 は、アプリケーションのうちネイティブ部分実行部

20

#### 【 0 0 3 9 】

ウェブブラウザ 4 0 は、ネットワーク 2 0 4 を介してサーバ 2 0 2 から得られた情報等を表示部 3 4 に画像表示する等の各種の機能を実現する。また、ウェブブラウザ 4 0 にはブラウザキャッシュ（ブラウザ記憶部）4 0 a が設けられている。ブラウザキャッシュ 4 0 a は、ウェブブラウザ 4 0 が各種処理を実行する際に使用される情報の記憶に利用される。ブラウザキャッシュ 4 0 a には、更新が頻繁に行われない情報を記憶しておくことが好適である。

30

#### 【 0 0 4 0 】

情報端末 2 0 0 は、サーバ 2 0 2 から情報を受信する際に、プロキシ 4 2 を介してサーバ 2 0 2 へアクセスする。すなわち、ネイティブ部分実行部の実行に必要な情報をサーバ 2 0 2 から取得する場合、及び、ウェブブラウザ 4 0 での処理に必要な情報をサーバ 2 0 2 から取得する場合のいずれにおいても、アプリケーションに含まれるネイティブ部分実行部によって実現されるプロキシ 4 2 を介して行われる。

#### 【 0 0 4 1 】

ウェブブラウザ 4 0 は、情報端末 2 0 0 での表示処理の負荷を低減するために、ネイティブキャッシュ 3 8 に記憶されているファイルの一部又は全部をブラウザキャッシュ 4 0 a に記憶しておくことができる。ウェブブラウザ 4 0 は、まずブラウザキャッシュ 4 0 a に必要な情報が記憶されているか否かを調べ、記憶されていればその情報を読み出して表示等の処理を行う。ブラウザキャッシュ 4 0 a に情報が記憶されていなければ、ウェブブラウザ 4 0 は、プロキシ 4 2 に情報を要求する。プロキシ 4 2 は、サーバ 2 0 2 に情報を要求する前にネイティブキャッシュ 3 8 に要求された情報が記憶されているか否かを確認し、記憶されていればその情報を読み出してウェブブラウザ 4 0 に受け渡す。ネイティブキャッシュ 3 8 に情報が記憶されていなければ、プロキシ 4 2 は、ダウンロード 4 4 によってサーバ 2 0 2 にその情報を要求する。

40

#### 【 0 0 4 2 】

50

ここで、上述したようにサーバ202は、アセットデータを含むファイルとそのファイルに対して算出されたハッシュ値を対応付けて記憶しており、情報端末200からアセットデータの要求を受けると、要求されたアセットデータを含むファイルを送る際に対応付けられたハッシュ値をメタデータとして付加して送信する。

#### 【0043】

情報端末200の処理部30は、受信したファイル（アセットデータ）およびハッシュ値をファイル名に関連付けてデータテーブル等としてネイティブキャッシュ38に記憶させる。アセットデータは、Base64等によりエンコードされてなくてもよく、バイナリのデータとして扱うことができる。

#### 【0044】

<データ取得手順>

以下、図3のフローチャートを参照して、ネットワーク204を介して情報端末200がサーバ202からアセットデータを取得する処理について説明する。本実施形態においては、ゲームで表示されるページ毎にウェブブラウザ40がサーバ202にアクセスし、ページを表示するためのアセットデータを取得するものとして説明を行う。

#### 【0045】

情報端末200においてゲームのアプリケーションが実行されると、ネットワーク204を介して情報端末200とサーバ202とが相互に情報の送受信が可能となるように接続される。ここで、図4に示すように、1つのページが情報端末200において表示されている状態から次のページの表示に移る際に、ウェブブラウザ40において新たな情報がウェブブラウザ40にて必要となる。

#### 【0046】

まず、ウェブブラウザ40は、画面表示等に必要なアセットデータがブラウザキャッシュ40aに記憶されているか否かを確認する。ブラウザキャッシュ40aには、更新がないアセットデータ等が記憶されており、必要なアセットデータがブラウザキャッシュ40aに記憶されていれば読み出して利用する。

#### 【0047】

ブラウザキャッシュ40aに必要なアセットデータが記憶されていなければ、ウェブブラウザ40は、プロキシ42にそのアセットデータを要求する。ウェブブラウザ40とプロキシ42との間では、情報はハイパー・テキスト・トランスファー・プロトコル（http）に則って行うことができる。

#### 【0048】

プロキシ42は、ネイティブ部分実行部によって実現されるダウンローダ44を用いて、要求されたアセットデータをさらにサーバ202に要求する。

#### 【0049】

サーバ202は、要求を受けたアセットデータを含むファイルのファイル名（file1, file2, ...）と対応するハッシュ値（バージョン情報）とを関連付けて情報端末200に送信する。たとえば、ファイル名"file1"とそれに対応付けられたハッシュ値hash1、ファイル名"file2"とそれに対応付けられたハッシュ値hash2、を含む場合、（<html> file1 hash, file2 hash, ...）のようなデータが送信される。

#### 【0050】

このとき、送信時にファイルのハッシュ値を計算し、算出されたハッシュ値をファイル名と対して送信することが好適である。これにより、送信時において最新のアセットデータに基づいて算出されたハッシュ値を確実に送信することができる。一方、ある程度のタイムラグが許容される場合、予めハッシュ値を算出し、メモリ20にファイルとハッシュ値とを関連付けてメモリ20に記憶しておき、送信時にはファイル名と対応するハッシュ値とを読み出して送信してもよい。この場合、サーバ202において送信毎にハッシュ値を算出する必要がなくなり、サーバ202の演算処理の負荷を低減することができる。

#### 【0051】

このように、次ページの表示に必要なアセットデータのファイルを送信する前に、ファ

10

20

30

40

50



イルの本体（内容）となるアセットデータは送信せず、ファイルに固有の識別情報であるファイル名と、そのバージョン情報であるハッシュ値が送信される。これが、プロキシ 4 2 において取得される。

【 0 0 5 2 】

プロキシ 4 2 は、受け取ったハッシュ値と同一のハッシュ値がネイティブキャッシュ 3 8 に記憶されているかを判定する。この判定結果において、同一のハッシュ値が記憶されていれば、そのハッシュ値に関連付けられたファイル名を有するファイルについては既に最新のアセットデータがネイティブキャッシュ 3 8 に記憶されていると判断できる。この場合、そのファイルのアセットデータはサーバ 2 0 2 から改めて取得する必要はない。そこで、プロキシ 4 2 は、サーバ 2 0 2 から該当するファイルを取得することなく、ネイティブキャッシュ 3 8 から該当ファイルのアセットデータを読み出し、読み出したアセットデータをウェブブラウザ 4 0 へ受け渡す。ウェブブラウザ 4 0 は、受け取ったアセットデータを用いて画像表示等の処理を行う。

10

【 0 0 5 3 】

なお、ハッシュ値の同一性を判定する際に、ファイル名の同一性も同時に判定してもよい。すなわち、プロキシ 4 2 は、受け取ったファイル名とハッシュ値との組と同一のファイル名とハッシュ値との組がネイティブキャッシュ 3 8 に記憶されているかを判定してよい。これにより、ファイル名で特定された同一のファイルのアセットデータについてハッシュ値を用いてデータの同一性を確認することができ、より確実に最新のアセットデータであるか否かを判定することができる。

20

【 0 0 5 4 】

図 3 の例は、ファイル file3 については同一のハッシュ値を有するファイルが既に登録されており、ファイル file1, file2 については同一のハッシュ値を有するファイルがなかった場合を示している。この場合、プロキシ 4 2 は、ダウンロード 4 4 を用いて 2 つのファイル file1, file2 のみを取得するリクエストをサーバ 2 0 2 へ送信する。この例では、A J A X (Asynchronous JavaScript + XML) を用い、\_get bundle file1, file2 のようなリクエストをサーバ 2 0 2 に送っている。

【 0 0 5 5 】

このリクエストに応じて、サーバ 2 0 2 は、該当するファイル file1, file2 をメモリ 2 0 から読み出し、これらファイルを情報端末 2 0 0 へ送信する。このときは、読み出したファイルに対してハッシュ値を計算して、ファイルと対応ハッシュ値をセットにして情報端末 2 0 0 のプロキシ 4 2 に提供する。すなわち、ファイル file1, file2 について、それぞれのハッシュ値 hash と、ファイルの本体（内容）となるアセットデータ body を含めてプロキシ 4 2 に送信する。このとき、アセットデータ body は、B a s e 6 4 等に則ってエンコードされていてもよいし、エンコードされていないバイナリデータとしてもよい。

30

【 0 0 5 6 】

ここで、サーバ 2 0 2 のデータ更新部 2 2 は、送信する段階でハッシュ値を計算して、その計算で得られたハッシュ値を送信することが好適である。これにより、送信されるハッシュ値はその時点の最新のアセットデータに対するハッシュ値となる。したがって、情報端末 2 0 0 において、記憶されるハッシュ値が最新のバージョンのアセットデータに対応したものであることが担保される。

40

【 0 0 5 7 】

一方、ある程度のタイムラグが許容される場合、予めハッシュ値を算出し、メモリ 2 0 にアセットデータとハッシュ値とを関連付けてメモリ 2 0 に記憶しておき、送信時にはアセットデータを含むファイルと対応するハッシュ値とを読み出して送信してもよい。この場合、サーバ 2 0 2 において送信毎にハッシュ値を算出する必要がなくなり、サーバ 2 0 2 の演算処理の負荷を低減することができる。

【 0 0 5 8 】

プロキシ 4 2 は、ファイル名、ファイル本体、ハッシュ値のうち少なくともファイル本体をウェブブラウザ 4 0 へ転送する。このとき、プロキシ 4 2 において、受信したアセッ

50

トデータのコンテンツ変換等の処理を行ってもよい。例えば、ウェブブラウザ40でサポートされているデータ形式等に変換する処理を行ってもよい。プロキシ42は、ネイティブ部分によって実現されているので、情報端末200の記憶部36の容量、表示部34の画面の解像度、情報端末200の処理部30の処理速度や利用できるデータ形式等の情報を情報端末200から得ることが容易であり、これらの情報に基づいて情報端末200毎に応じた処理を適用できる利点がある。

【0059】

ウェブブラウザ40は、受け取ったアセットデータを用いて画像表示等の処理を行う。また、ウェブブラウザ40は、必要に応じて受け取ったアセットデータをブラウザキャッシュ40aに記憶（キャッシング）させる。

10

【0060】

また、プロキシ42は、ファイル名、ファイル本体、ハッシュ値をネイティブキャッシュ38に記憶させる。このような処理を行い、プロキシ42は、ネイティブキャッシュ38の記憶完了を受け、サーバ202に受け取り通知を送信する。

【0061】

このように、ユーザが情報端末200を使用してゲームについての一定の操作を行う都度、情報端末200で実行されるアプリケーションに応じてプロキシ42（ダウンロード44）を用いてウェブブラウザ40で使用するアセットデータをサーバ202から取得することができる。

【0062】

20

このとき、サーバ202からページの表示に必要なファイルのデータがそのまま送られて来るのではなく、ファイル名とそのハッシュ値がまず送られてくる。そこで、そのハッシュ値を利用して、当該ファイル（ファイル本体）がネイティブキャッシュ38に記憶されているのかを判定し、記憶されている場合にはサーバ202に、ファイル本体の送信を要求しない。これによって、通信するデータ量を大幅に削減することができ、通信の効率を高めることができる。したがって、ネットワーク204の通信容量が十分でない場合であっても必要なデータのみを通信するようにすることができる。

【0063】

また、ファイルの本体であるアセットデータ自体はプロキシ42からアクセスされるネイティブキャッシュ38に記憶することで、ブラウザキャッシュ40aのみを用いる場合に比して、データ容量を気にせずにファイルをキャッシュすることができる。一般的に、ウェブブラウザ40からアクセス可能なブラウザキャッシュ40aは容量が小さく、アセットデータ、特に画像データやトゥルータイプフォント（ttf）等の容量の大きなアセットデータのキャッシングには適していない。一方、ネイティブキャッシュ38は、ネイティブアプリケーションによって実現されるプロキシ42からアクセス可能であり、情報端末200の記憶部36の容量等に応じて比較的自由に容量を設定できるので、画像データやトゥルータイプフォント（ttf）等の容量の大きなアセットデータのキャッシングには適している。

30

【0064】

また、ウェブブラウザ40からみれば、従来のようにサーバ202にアクセスすると同様にプロキシ42にアクセスすることによってアセットデータを取得することができる。プロキシ42では、ネイティブキャッシュ38のアクセスが許容され、ネイティブキャッシュ38に記憶されたデータについて破損がないか等のチェックをウェブブラウザ40から独立して実行すること等が可能であり、その他の利用も可能となる。

40

【0065】

さらに、アプリケーションにプロキシ42及びダウンロード44の機能を実現するネイティブ部分実行部を組み込むことによって、アプリケーションによる処理が終了した時点でプロキシ42及びダウンロード44の機能も終了させることができる。これにより、ゲーム等の処理を終了させた時点でプロキシ42やダウンロード44の処理も終了させることができ、情報端末200のリソースを開放して他のアプリケーションでの利用を可能と

50

したり、情報端末 200 のバッテリーの消耗を低減させたりすることができる。

【0066】

なお、本実施の形態では、1 ページの表示毎にアセットデータを要求する例について説明したが、これに限定されるものでなく、複数のページの表示に必要なアセットデータを一度に纏めて取得してもよい。

【0067】

また、情報端末 200 からのアクセス時において、それがそのゲームについての初回のアクセスであることを判定する手段を有すれば、その最初の応答では、送信すべきデータを J S O N (JavaScript Object Notation) 化してハッシュ値とデータを一度に転送してもよい。

10

【0068】

また、サーバ 202 において、ハッシュ値を算出中にクエリがあると、サーバ 202 は、古いハッシュ値を返す場合があることも想定される。たとえば、ファイルのバージョンについて、ファイルシステムに依存しないフィンガープリントを用いているならば、

(1) 準備用ディレクトリツリーに新規ファイルを置く

(2) 追加ファイルのハッシュ値を算出する

(3) ハッシュ値が算出できたところで、リリース用ディレクトリツリーに置く  
こととしてもよい。

【0069】

また、複数ファイルを不可分に更新する場合への対応として上記のような対応をすることも可能である。

20

【0070】

さらに、上記処理の中で、ブラウザキャッシュ 40 a に記憶するデータと、ネイティブキャッシュ 38 に記憶するデータを最適化してもよい。

【0071】

<変形例>

上記実施の形態では、情報端末 200 はネットワーク 204 を介してサーバ 202 から情報を直接受信する態様としたが、これに限定されるものではない。本変形例では、情報端末 200 は、コンテンツ・デリバリ・ネットワーク (C D N) 206 を介してサーバ 202 から情報を受信する。C D N 206 は、コンテンツ配信などを行うサイト運営側がエンドユーザーに対して効率よく安定して提供するネットワークであり、アクセス集中等によるサーバ 202 の負担を軽減することを可能とする。

30

【0072】

C D N 206 は、サーバ 202 のアセットデータと同期をとっているサーバ (以下、キャッシュサーバ) を複数台接続し、情報端末 200 からのアクセスをキャッシュサーバに振り分ける構成とされる。これにより、サーバ 202 へのアクセスを分散させることができる。C D N 206 で構成されたサイトへアクセスする場合、情報端末 200 は初めにサーバ 202 にアクセスする。しかしながら、そこでデータのやり取りは行われず、サーバ 202 が情報端末 200 から最も近くにある C D N 206 のキャッシュサーバにアクセスを振り分ける。情報端末 200 からのこれ以降のアクセスは、振り分けられたキャッシュサーバに対して行われるように設定され、データの要求に対するレスポンスも良くなり、サーバ 202 の負荷も軽減される。

40

【0073】

<実施形態の効果>

このように、本実施形態では、以下のような効果が得られる。

・ネイティブキャッシュ 38 においてファイルの本体であるアセットデータを記憶できるため、ブラウザキャッシュ 40 a の容量が小さくても問題が少ない。

・ブラウザキャッシュ 40 a はウェブブラウザ 40 により利用されるものであり、ウェブブラウザ 40 の外にある処理部 30 などによっては制御が難しい。本実施形態では、ネイティブ部分実行部ではネイティブキャッシュ 38 に記憶されているアセットデータを用い

50

ることができるので、アプリケーションからブラウザキャッシュ４０aに記憶されているデータを扱う必要がなくなり、そのような問題は生じない。ウェブブラウザ４０からはプロキシ４２を介してネイティブキャッシュ３８に記憶されているデータを用いることができるので効率的な処理が行え、サーバ２０２との間で無駄な通信が発生することを防止できる。

・ゲーム等のアプリケーション起動時は、アプリのダウンロードだけを必要とし、大きな容量のデータのダウンロードなど特別な処理を必要とせず、起動時間を短縮できる。

・アプリケーション起動中でも、アセットデータなどのアップデートが可能であり、ユーザが必要とする時点での最新データを供給できる。

・アプリケーションによる処理を終了させた時点でプロキシ４２やダウンロード４４の処理も終了させることができ、情報端末２００のリソースを開放して他のアプリケーションでの利用を可能としたり、情報端末２００のバッテリーの消耗を低減させたりすることができる。

10

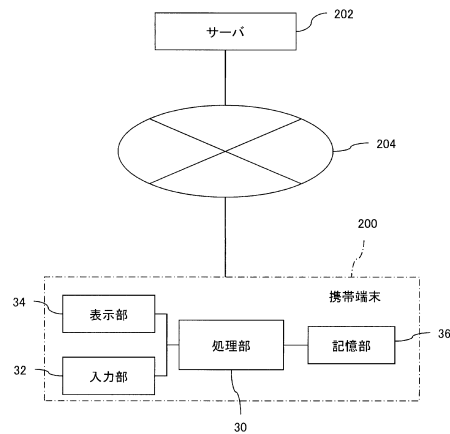
# 【符号の説明】

## 【００７４】

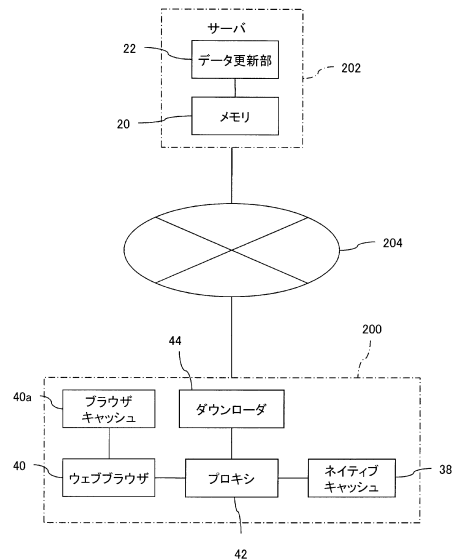
１０ ウェブブラウザ、１０a ダウンローダ、１０b ブラウザキャッシュ、１２ ネイティブ、１２a ストレージ、２０ メモリ、２２ データ更新部、３０ 処理部、３２ 入力部、３４ 表示部、３６ 記憶部、３８ ネイティブキャッシュ、４０ ウェブブラウザ、４０a ブラウザキャッシュ、４２ プロキシ、４４ ダウンローダ、１００ 携帯端末、１０２ サーバ、１０４ ネットワーク、２００ 情報端末、２０２ サーバ、２０４ ネットワーク。

20

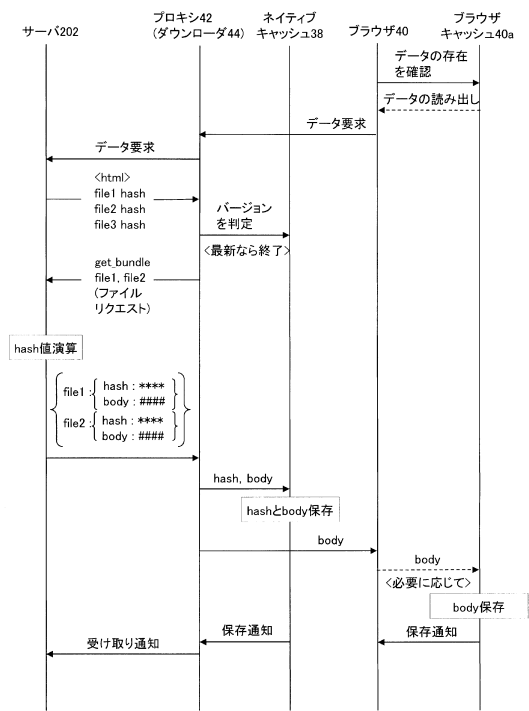
### 【図１】



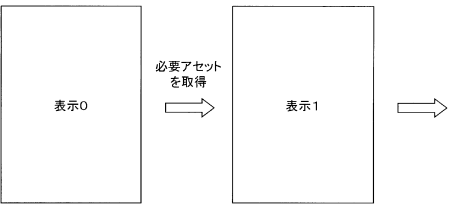
### 【図２】



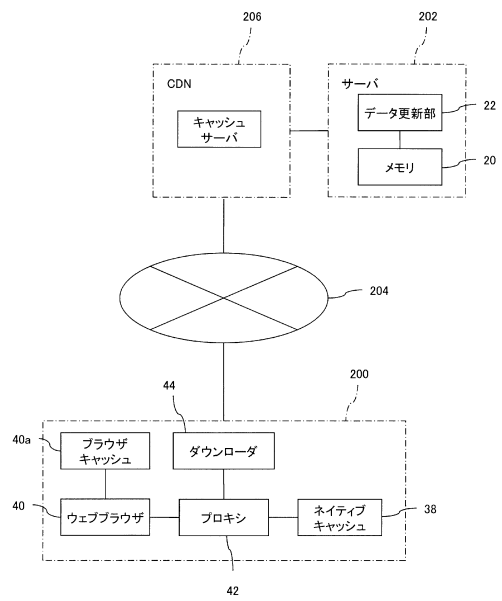
【図 3】



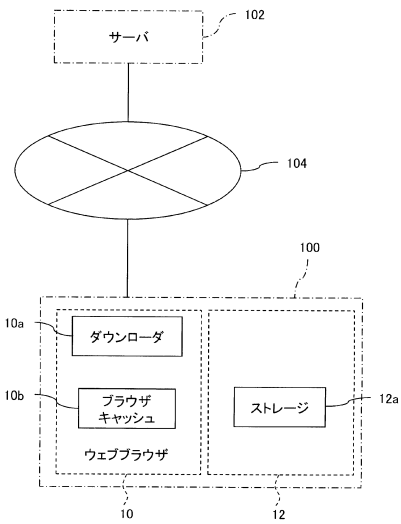
【図 4】



【図 5】



【図 6】



---

フロントページの続き

特許法第30条第2項適用 2013年6月24日にGoogle Play: [https://play.google.com/store/apps/details?id=kr.daum\\_mobage.am\\_db.g13000622](https://play.google.com/store/apps/details?id=kr.daum_mobage.am_db.g13000622) iOS: <https://itunes.apple.com/kr/app/doteuhieoloji/id655006139?ls=1&mt=8>において、ソフトウェア「 」(ドットヒーローズ)を販売した。

(56)参考文献 米国特許出願公開第2002/0143896(US, A1)

特開2011-180820(JP, A)

特開2001-016202(JP, A)

(58)調査した分野(Int.Cl., DB名)

G06F 13/00

G06F 12/00