



#### (6) 画面描画

画面上にアプリケーションの初期画面を描画する区間。アプリケーション/X サーバ/ウィンドウマネージャのスイッチが頻繁に起こる。

### 3. 考察

測定結果では、起動に 2.34 秒程度掛かることが判明した。全体処理の内、実際の描画処理は全体の 27%であり、また、キー入力からアプリケーションの起動までは 4%程度であるため、結果としてアプリケーションの初期化処理に全体の 70%近く消費していることがわかる。またアプリケーションの初期化処理の内、共有ライブラリなど OS レベルでの初期化処理は 8%程度であり、残りの 61%は GUI レベルでの初期化処理であることが判明した。

プロセッサ能力や XIP による ROM/RAM 実行性能差などの差異はあるものの、X/GTK を利用した場合、各処理の時間消費率はほぼ同様になると考えられ、起動に 1 秒以上掛かることが予想される。これはコンシューマ機器では致命的な問題である。

アプリケーションの初期化処理を削減する方法の 1 つとして、予め必要なアプリケーションを起動しておき、キー入力が入ると、選択されたアプリケーションの画面に切り替えるという方法が考えられる。しかし、この場合ではアプリケーションを起動することによるメモリ使用量の増加が問題となる。XIP 化によって、メモリ削減は期待できるもの、データ/ヒープ/スタックはメモリに配置するため、メモリ消費量は増加(測定では 2MB 程度増加)し、メモリ資源が乏しいコンシューマ機器には推奨できない。よって、通常の起動による改善策を検討する必要がある。

### 4. 改善案検討

今回の測定結果を元に、改善が可能な箇所について検討した。以下、検討した内容について述べる。

#### 4.1 OS レベルでの改善

##### (1) ファイル読み出しの高速化

ウィンドウ生成処理(図 3)においてビットマップファイルの読み出しに全体の 42%が消費されている。これらファイルは処理の重い JFFS2 に置かれており、read()に時間が掛かっている。そこで、これらファイルの置き場所をより高速なファイルシステムに変更する。例えば、ファイルイメージを ROM 化し、フラッシュメモリから直接参照できるようなファイルシステムを構築すれば、この読み出し時間をほぼ 0 にすることができる。コンシューマ機器用アプリケーションでは、メニュー画面等で使用されるビットマップファイルは基本的に固定であり、より簡易なファイルシステムでも対応できると考える。

##### (2) 共有ライブラリアドレス検索処理の高速化

共有ライブラリは、アプリケーション実行時に必要となるライブラリがロードされ、アプリケーション内で未定義になっているライブラリ関数のリンク処理が行われる。特に GUI を利用するアプリケーションは大量の X/GTK ライブラリをリンクするため、リンク処理に時間が掛かる。家電などのコンシューマ機器では、使用するアプリケーションは決まっており、使用する共有ライブラリも固定的である。よって、プロセス空間上で各共有ライブラリ位置を固定化し、そのアドレス情報をアプリケーションの共有ライブラリリンク情報領域に予め格納すればこのリンク処理を省くことが可能となると考える。

#### 4.1 GUI ライブラリレベルでの改善

##### (1) ロケール処理の共有化

ロケール処理では、各種言語用にロケール情報からロケールデータを作成するため、処理に時間が掛かっている。基本的に言語は出荷時に固定的であり、またアプリケーション毎で異なることはないと考えられる。よって、出荷時に予め作成したロケールデータを、各アプリケーションで共有化しておけば、アプリケーション毎のロケール処理を省くことが可能となると考える。

##### (2) フォント処理の共有化

ウィンドウ生成処理(図 3)では、フォントの初期化も行われている。これもロケール処理と同様にフォント情報からフォントデータをアプリケーション毎に生成するため、時間が掛かる。フォントも各アプリケーションで共通に利用することが可能であり、フォント処理を全プロセスで共有化することで、各アプリケーション毎に行っているフォント初期化処理を削除することが可能と考える。

#### 4.3 改善効果

上記改善策を施した場合の効果を図 3 に示す。前述した改善策が実現できれば、41%の削減効果が期待でき、処理時間を 2/3 まで短縮できる可能性がある。

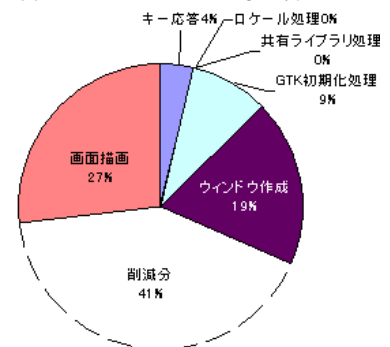


図 3: 改善策の効果

#### 5. おわりに

本稿では、コンシューマ向け組込み機器に対し、組込み Linux が適用可能かという観点のもと、GUI を備えたアプリケーション起動性能の測定結果と、起動の改善策について述べた。評価の結果、全体の 70%近くがアプリケーションの初期化処理に費やされ、かつ 61%が GUI レベルでの初期化処理であることが判明した。またこれら初期化処理に幾つかの改善を施すことで、2/3 まで起動処理を短縮できる可能性があることを述べた。

今回検討した改善策だけでは、依然として起動時間は 1 秒を切れておらず、さらなる起動時間短縮が必要である。その方策の 1 つとして XIP をやめることが考えらる。実際に測定した結果では XIP を実施しないほうが起動性能が向上する結果がでていた。これにより実行性能が 1.6 倍(RAM/フラッシュの性能差 3 倍)になるため、これを見込めば起動時間が 1 秒以下になる可能性がある。しかし、XIP をやめることは、メモリリソースを消費することになり、搭載できる RAM 量の問題から採用すべきか判断できていない。

今後は、これら改善策の検証を行うとともに、XIP の効果についてもより詳細に検証していく予定である。

#### 参考文献

- [1] 町田、田中、篠原、大和田、水山、天野、関口：  
「Linux をデジタル家電向けに改良、起動時間と応答時間を半分に」、日経エレクトロニクス 2003.7.7 号、p.131 ~ 142
- [2] 丸山、八木、森田、吉本：  
「情報家電における OS 起動時間解析」、情報処理学会第 66 回全国大会(2004)
- [3] CE-Linux Forum 1.0 Specification：  
[http://www.celinuxforum.org/PublicSpecifications/CELF\\_Specification\\_V\\_1\\_0\\_R2.pdf](http://www.celinuxforum.org/PublicSpecifications/CELF_Specification_V_1_0_R2.pdf)