

OS カーネル作成処理における ページ例外に着目した OFF2F の効果予測

額田 哲彰† 佐藤 将也‡ 谷口 秀夫‡
† 岡山大学工学部 ‡ 岡山大学大学院自然科学研究科

1 はじめに

ハードウェア技術の進歩により、バイトアクセス可能な不揮発性メモリ (Non-Volatile Memory : 以降, NVメモリ) が登場している。また、揮発性メモリと NVメモリ を混載した環境における新たな実行ファイル形式 (OFF2F : Object File Format consisting of 2 Files) が提案されている [1]。本稿では, FreeBSD 11.0-RELEASE (以降, FreeBSD) の OS カーネル作成処理において, OFF2F を適用することによるページ例外処理時間の短縮効果を示す。

2 OFF2F

実行ファイルは, 4つの内容 (テキスト部, ヘッダ部, データ部, および関係情報部) から構成されている。OFF2F は, 実行ファイルが2つのファイルに分割されている形式である。具体的には, 読み込みのみ行われるテキスト部のみのファイル, およびヘッダ部とデータ部と関係情報部を一つにまとめたファイルである。

OFF2F プログラム実行時のマッピングの様子を図1に示す。テキスト部に対するページ例外発生時には, NVメモリの該当ページを仮想空間にマッピングする。これにより, 外部記憶装置から実メモリへの読み込みを削減し, プログラム実行を高速化できる。

3 OS カーネル作成処理の短縮効果

3.1 観点

OS カーネル作成処理におけるページ例外処理時間を評価し, OFF2Fの有効性を述べる。具体的には, make コマンドを用いた FreeBSD カーネル作成時に発生するページ例外回数をもとに, OFF2F を適用した際の処理時間の短縮効果を予測する。

3.2 OS カーネル作成処理におけるプログラム

OS カーネル作成処理で実行されるプログラムのテキスト部とデータ部のページ数, および各プログラムが fork() する回数と exec() で起動される回数を表1に示す。また, プログラムにリンクされるライブラリのテキスト部とデータ部のページ数, および各ライブラリをリンクするプログラム数を表2に示す。なお, ページサイズは 4 KB である。

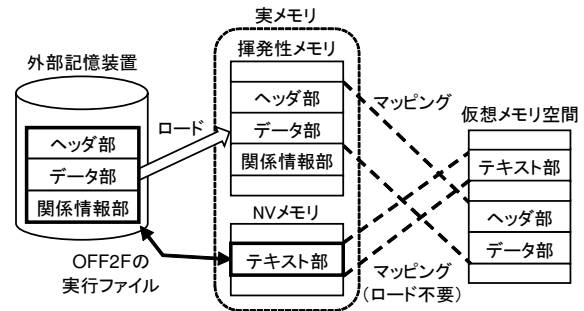


図1 OFF2F プログラム実行時のマッピング

3.3 ページ例外数

ページ例外が発生する回数は, プログラムの fork() 回数や exec() 回数, およびライブラリをリンクするプログラムの fork() 回数や exec() 回数に依存する。したがって, 各部は以下に依存する。

- (1) プログラムのテキスト部 : ページ数 × exec() で起動される回数
- (2) プログラムのデータ部 : ページ数 × (fork() する回数と exec() で起動される回数の和)
- (3) ライブラリのテキスト部 : ページ数 × リンクするプログラムが exec() で起動される回数
- (4) ライブラリのデータ部 : ページ数 × (リンクするプログラムが fork() する回数と exec() で起動される回数の和)

ここで, 上記の数値は, そのプログラム全体またはライブラリ全体が一度だけ読まれたときに発生するページ例外の回数 (以降, 全 PF 数) と解釈できる。

3.4 評価

文献 [2] と同様の評価環境を想定する。つまり, 以下のように各処理の時間を定義する。

- t_1 : 実メモリ確保 (1 ページ)
 - t_2 : 外部記憶装置からデータ (4KB) を読み込む
 - t_3 : 実メモリをマッピング表に登録
 - t_4 : NVメモリのページをマッピング表に登録
- また, 各処理の時間は, $t_2 = 0.092 \text{ ms}$ (DK), 0.017 ms (SSD), $t_i = 0.001 \text{ ms}$ とする。外部記憶装置からのデータの読み込みは, 連続読み込みを仮定している。これは, OS カーネル作成処理で用いられるプログラムは呼び出し回数が多いためである。さらに,
- P : プログラムにおける (テキスト部 + データ部) が占めるページ数
 - S : プログラムに占めるテキスト部の割合

Effect Prediction of OFF2F Focused on Page Fault Handling in Building OS Kernel.
Tetsuaki Nukata†, Masaya Sato‡, Hideo Taniguchi‡
†Faculty of Engineering, Okayama University
‡Graduate School of Natural Science and Technology, Okayama University

表1 OS カーネル作成処理におけるプログラム

通番	プログラム名	テキスト部のページ数	データ部のページ数	fork()する回数	exec()で起動される回数	テキスト部の全PF数	データ部の全PF数
1	as	362	5	0	20	7,240	100
2	awk	48	1	858	843	40,464	1,701
3	cat	3	1	0	10	30	10
4	cc	10,699	5	4,577	9,132	97,703,268	68,545
5	chmod	2	1	0	1	2	1
6	cp	4	1	0	2	8	2
7	ctfconvert	25	1	0	18,200	455,000	18,200
8	ctfmerge	17	1	0	3,268	55,556	3,268
9	date	5	1	0	1	5	1
10	dirname	1	1	0	1	1	1
11	echo	1	1	0	256	256	256
12	find	12	1	0	3	36	3
13	grep	24	1	0	13	312	13
14	hostname	1	1	0	1	1	1
15	ld	391	6	0	3,397	1,328,227	20,382
16	ln	2	1	0	5	10	5
17	make	173	4	17,781	1,735	300,155	78,064
18	mv	3	1	0	44	132	44
19	nm	23	1	0	815	18,745	815
20	objcopy	27	2	0	6,015	162,405	12,030
21	realpath	1	1	0	1	1	1
22	rm	3	1	0	9	27	9
23	sed	9	1	0	7	63	7
24	sh	36	1	1,747	11,532	415,152	13,279
25	size	4	1	0	1	4	1
26	sort	14	1	0	1	14	1
27	svnliteversion	391	1	0	2	782	2
28	touch	2	1	0	1	2	1
29	uudecode	3	1	0	132	396	132
30	xargs	4	1	481	807	3,228	1,288
	合計	12,290	47	25,444	56,255	100,491,522	218,163

表2 プログラムにリンクされるライブラリ

通番	ライブラリ名	テキスト部のページ数	データ部のページ数	リンクするプログラム数	テキスト部の全PF数	データ部の全PF数
1	libarchive.so.6	178	4	1	1,070,670	24,060
2	libbsdxml.so.4	37	2	2	222,629	12,034
3	libbz2.so.4	19	2	2	114,532	12,056
4	libc.so.7	398	12	26	16,704,458	540,684
5	libcrypto.so.8	575	41	1	3,458,625	246,615
6	libdwarf.so.4	44	1	2	836,660	19,015
7	libedit.so.7	52	2	1	599,664	26,558
8	libelf.so.2	22	1	5	622,578	28,299
9	libgnuregex.so.5	21	1	1	273	13
10	liblzm3.so.5	40	1	1	240,600	6,015
11	libm.so.5	42	1	1	35,406	1,701
12	libmd.so.6	23	1	1	23	1
13	libncursesw.so.8	87	5	1	1,003,284	66,395
14	libprivatesqlite3.so.0	302	4	1	604	8
15	libthr.so.3	27	1	4	742,095	27,485
16	libz.so.6	23	1	5	632,454	27,498
	合計	1,890	80	55	26,284,555	1,038,437

表3 ページ例外処理時間 (s)

	テキスト部のページ例外処理時間	データ部のページ例外処理時間	合計
従来 (DK)	11,917	118	12,035
OFF2F (DK)	127	118	245
従来 (SSD)	2,409	24	2,433
OFF2F (SSD)	127	24	151

と定義し、従来のページ例外処理（以降、従来方式）とOFF2Fを適用した方式におけるページ例外処理の時間を、それぞれ以下の式(1)、(2)で定式化する。

$$(t_1 + t_2 + t_3)PS + (t_1 + t_2 + t_3)P(1 - S) \quad (1)$$

$$t_4PS + (t_1 + t_2 + t_3)P(1 - S) \quad (2)$$

まず、各プログラムおよび各ライブラリそれぞれのテキスト部の全PF数とデータ部の全PF数について、3.3節の(1)から(4)を用いて算出する。例えば、ライブラリのlibarchive.so.6のテキスト部の全PF数を算出する。libarchive.so.6を調査したところ、リンクするプログラムはobjcopyのみであった。このため、表1、表2、および3.3節の(3)より、libarchive.so.6のテキスト部の全PF数は $178 \times 6,015 = 1,070,670$ 回である。同様な方式で算出した各全PF数を表1と表2に示す。

次に、ページ例外処理時間について、テキスト部とデータ部のページ例外処理時間を算出する。例えば、従来(DK)場合、式(1)の第1項より、テキスト部のページ例外処理時間は $(t_1 + t_2 + t_3)PS$ であり、11,917sである。同様な方式で導出した各処理時間を表3に示す。

表3より、従来のページ例外処理時間全体のうち、約

99%がテキスト部のページ例外処理時間である。OFF2Fでは、テキスト部の読み込みを省略できることから、ページ例外処理をDKで約49倍(12,035/245)、SSDで約16倍(2,433/151)高速化できる。

4 おわりに

FreeBSDのOSカーネル作成処理にOFF2Fを適用することによる処理時間の短縮効果を示した。OFF2Fは従来に比べてOSカーネル作成処理をDKでは約46倍、SSDでは約16倍高速化できることを示した。

残された課題として、テキスト部とデータ部が読み込まれる割合を変化させた場合の評価がある。

謝辞 本研究の一部は、JSPS KAKENHI 18K11244、および共同研究(株式会社富士通研究所)による。

参考文献

- [1] 谷口 秀夫：揮発/不揮発メモリ混載環境を支援する仮想記憶機構向け実行ファイル形式：OFF2Fの提案，コンピュータシステム・シンポジウム論文集，Vol.2017，pp.35-40 (2017.12)。
- [2] 谷口 秀夫，佐藤 将也，河辺 誠弥，横山 和俊：CoW機能を考慮したOFF2Fプログラムのページ例外処理の評価，情報処理学会論文誌 (2020) (掲載予定)。