B-020

NetBSD仮想メモリ予約機構の実装と評価

Implementation and Evaluation of Virtual Memory Reservation Mechanism for NetBSD Operating System

狩野秀一[1]
Shuichi Karino

NECシステムプラットフォーム研究所[1]
System Platforms Research Laboratories, NEC Corporation.

1 まえがき

NetBSD[1]は仮想メモリ予約機構を持たないため，プロセスは実際に利用できる量より多くの仮想メモリを確保できる．同OSはスワップを持たない組み込みシステムでもしばしば利用されるが，そのような環境では実メモリ不足によるプロセス異常終了のリスクが高くなる．本稿では，組み込みシステム向けにNetBSD上に実装したメモリ予約機構の概要を示し，同実装をある組み込みシステムに適用した結果を示す．

2 組み込みシステムにおけるメモリのovercommit問題

NetBSDは多くのプラットフォームをサポートすることを謳っている．その中には多くの組み込みシステム向のプラットフォームも含まれる．組み込みシステムでは，ハードディスク等の二次記憶装置を搭載できない場合も多いため，スワップ領域を持たない構成もしばしば採用される．スワップ領域は，ファイルシステム上に実体を持たない，ヒープやスタックなどの用途のメモリ（匿名メモリ）の待避場所として使われる．スワップ領域なしのシステムでは，匿名メモリの待避場所がないことになる．

NetBSDの仮想記憶実装であるUVMは，仮想メモリを確保するさいに，実メモリとスワップ領域の空きが十分かどうかを検査しない．このため，匿名メモリのための仮想メモリを，実際に利用可能な量よりも多く確保できる．この振る舞いにより，確保した匿名メモリを実際に利用する時に，実メモリやスワップ領域が不足して処理の継続ができなくなる問題が，メモリovercommit問題として指摘されていた[2]．広大なスワップ領域を搭載できるシステムでは実際にこの問題が起きることは稀だが，組み込みシステムでは，上記のように匿名メモリは実メモリ上に保持しなければならない場合があり，また，実装された実メモリの量にも制約がある場合も多いことから，overcommitが起こりやすい．

メモリ不足により匿名メモリを保持するための実メモリが確保できないと，その時点で走行しているプロセスにKILLシグナルが配送され，強制的に終了させられる．実メモリが枯渇する時点で走行しているプロセスを予測することは困難なため，メモリが不足するとシステムの安定した運用が困難になる．これを回避するには，匿名仮想メモリの量が，利用できる実メモリを越えないよう，メモリ確保量を予約する機構が必要になる．

3 匿名仮想メモリ予約方式の検討

匿名仮想メモリを予約する方式は，プロセスのリソースリミットを利用するものと，システムコールの度に匿名メモリの確保量を計算する機構を追加するものとがある．

前者は，プロセス毎のスタックおよびBSSの利用量にリソースリミットにより上限を設け，全プロセスのリソースリミットが空き実メモリの量に収まるように調整する．本方式はユーザー空間のみで実現できるが，事前にプロセス毎の最大メモリ使用量をリソースリミットに設定する必要があり，予約量の調整に手間がかかる．また，多くのプログラムはリソースリミットを動的に変更するようには実装されいないことから，動的なリソースの使い回しは困難であり，効率的なリソースの利用が難しい．

後者の方式は，システムコール呼出し時に，プロセスが要求したメモリ量から必要な匿名メモリ量を計算し，カーネル内で保持する予約総量に加算する．もし，予約量が空きメモリの量を越えていれば，システムコールの実行を失敗させる．この方式は，カーネルにて匿名メモリの予約量を一元的に管理するため，効率的にメモリを利用できる（図1）．ただし，カーネル内のシステムコールのコードへのリソース予約機構の追加が必要になる．本研究では，匿名メモリのより効率的な利用を目指して，後者の方式を採用する．

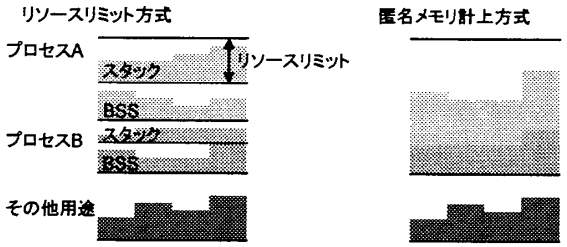

図1 メモリ予約2方式の比較

4 匿名仮想メモリ予約機構の実装

NetBSD 1.6上に匿名仮想メモリ予約機構を実装した．匿名メモリを確保／解放するシステムコールとスタックの伸長を行うトラップ処理に，匿名メモリ量の増減を計算するコードを追加した．システム全体での匿名メモリ量を合計する変数を用意し，匿名メモリを確保／解放する度に，同変数の値を増減するようにした．

匿名メモリの予約量は，システム全体の値のほか，増減の管理のためにプロセス毎にも保持するようにした．具体的には，プロセス毎に用意されているvmspace構造体のなかで，スタック及びBSSのサイズ保持に利用されているvm_ssize, vm_dsizeフィールドの値を，各々スタック及びBSSの予約値として利用することとし，さらにmmapされた匿名メモリの量としてvm_msizeフィールドを追加した．

各システムコール等の予約処理の概要は次の通りである．本予約機構を実装したシステムコールの一覧と実装場所を表 1 に示す．

システムコール	実装場所
fork	kern/kern_fork.c:fork1()
execve	kern/kern_exec.c:sys_execve()
brk	uvm/uvm_unix.c:sys_obreak()
mmap	uvm/uvm_mmap.c:sys_mmap()
munmap	uvm/uvm_mmap.c:sys_munmmap()
exit	uvm/uvm_glue.c:uvm_exit()

表 1　本予約機構を実装したシステムコール

fork 親プロセスが複製されるため，親プロセスと同じだけのスタック，BSS，mmap 用匿名メモリの量を新たに予約する．

execve 実行ファイルから取得できる BSS とスタックの初期値を予約すると共に，もともと走行しているプロセスの予約している全匿名メモリの量を解放する．

brk 伸長する BSS 分の匿名メモリの量を予約する．

mmap MAP_ANON フラグがセットされており，書き込み可能な場合のみ，map されるメモリ量を予約する．MAP_SHARED フラグがセットされているときは，fork しても予約が重複しないよう，別変数に計上するようにした．

munmap unmap される map entry を走査し，匿名メモリで，かつ書き込み可能なエントリのみについて，予約量を解放する．

exit プロセスが予約している全匿名メモリの量を解放する．

スタック スタックが伸長するときに発生するページフォルト処理に伸長量を予約するよう実装した．ただし，伸長に失敗するとプロセスの実行が困難になるため，fork を行うときにリソースリミットまで予約する方式も併用できるようにした．

上記のうち，スタック以外のシステムコール処理では，予約が失敗するとコード ENOMEM でエラーを返すようにした．これにより，匿名メモリが十分でない場合プロセスは自発的に異常系の処理を行うことができ，KILL シグナルを受信する場合と比較してより信頼性の高い処理が可能になる．

プロセスの匿名メモリに利用できるメモリ量は次のように計算した．カーネルが予約しているメモリと他のキャッシュ等に利用されるメモリの総量を計算し，実メモリ量からその量を差し引くことで空きメモリ量を算出した．具体的には，バッファキャッシュの最大値として変数 filemin および filemax の大きい方の値を利用した．実行コードキャッシュについても同様に execmin, execmax より計算した．また，カーネルが確保する submap 領域はその全体を予約した．その他 kernel map からとられるメモリは，確保が行われる都度予約した．これらの使用量を計算して，全物理ページ数から差し引くことで，予約可能な匿名メモリ量を算出した．

5　組み込みシステムへの適用

上記予約機構をある組み込みシステムに適用し，その有効性を検証した．同組み込みシステムの諸元は次の通りである．

- i386 アーキテクチャ
- RAM 256Mbytes（うち約 120Mbytes を memorydisk rootfs に利用）
- スワップ領域なし

NetBSD 標準のプロセスに加え，装置が提供するサービスや装置管理のためのプロセスが 50 個程度実行される．

上記のシステムについて本予約機構を適用したところ，起動シーケンスの途中でメモリ予約に失敗し，システムが立ち上がらなくなった．これは，実際に利用可能な量よりも多くの匿名メモリの確保を行おうとしたためで，システムが長時間運用される前にメモリ不足の可能性を検知できたことになる．

ユーザー空間のメモリ確保が始まってから，起動シーケンスが終了して平常の状態になるまでの期間の，匿名メモリの予約量と匿名メモリのために利用される物理ページ数を，予約失敗の動作を抑制して測定した．また，PC にインストールした NetBSD についても，起動してから X サーバ，emacs, mozilla を稼動させるまでについて同様の測定を行った．結果を図 2 に示す．横軸は，システムコール発行などのメモリ確保イベントの累積回数，縦軸はページ数を示す．

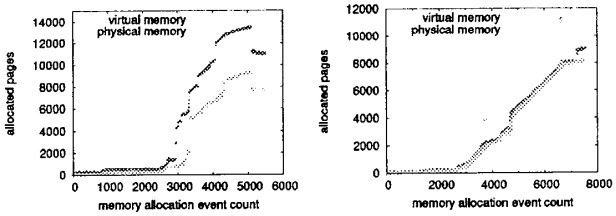

図 2　匿名メモリ確保の様子: 組み込みシステム（左）PC（右）

これらより，組み込みシステムでは，確保されている実メモリに比べ，予約されている仮想メモリの量が相当に多いことがわかる．調査の結果，これは，装置が保持すべき設定情報などの格納領域を装置仕様の値まであらかじめ確保している等のためであることがわかった．また，上記両システムとも，予約量が一時的に大きな値になっているが，実メモリの使用量がそこまで達しない場合があることもわかる．これらより，本予約機構を有効に利用するには，かなり効率的なメモリ利用が必要なことがわかる．

6　むすび

本稿では，NetBSD に実装した匿名仮想メモリ予約機構について概要を述べ，組み込みシステムに適用した結果を示した．本予約機構を導入することで，メモリ予約量の把握が可能になり，システムの信頼性を高めることができるようになった．ただし，粗雑なメモリ確保を行うと，未使用のメモリを多量に予約してしまうことで，システムの運用に大きな制約がかかることもわかった．

参考文献

[1] NetBSD Project. http://www.netbsd.org/

[2] みのうらまこと, “UNIX における仮想メモリシステムの設計と実装, ” BSD Magazine, No.4, pp.37-50, アスキー, 東京, 2000.