令和元年度 特別研究報告書要旨

非同期システムコール処理機構の提案

T160395　　中島 健

指導教員： 芝 公仁 助教，三好 力 教授

1　はじめに

ユーザプロセスがカーネルの機能を利用するためにはシステムコールを使用する．プロセスはシステムコールの実行完了を待つ必要がある．また，システムコール発行にはカーネルへ処理を移行するため，速度の低下が考えられる [1]．本稿では非同期システムコール処理機構について述べる．本機構により，システムコールの処理を待つことなくプロセスの処理が行える．

2　システム構成

本機構はカーネルでシステムコールの処理を行う exec_thread を作成し，exec_thread はシステムコールの処理を行う．システムコールを発行するスレッドと発行された要求を処理するスレッドをを分け，非同期でシステムコールを処理することが可能となる．プロセスは発行したいシステムコールの番号や引数をシステムコールリクエストとして扱う．プロセスはシステムコールリクエストを作成することで exec_thread に処理を依頼する．カーネルからリクエストへアクセスするためにはシステムコールを使用する必要がある．本機構にはオーバヘッドを削減するためにリクエスト領域と呼ばれるメモリ領域を作成し，システムコールを利用せずリクエストへのアクセスが可能となる．exec_thread はシステムコールリクエストの状態を確認し，要求があればリクエストの番号と引数を用いてシステムコール関数を呼び出す．システムコールの結果はリクエストに格納される．プロセスはリクエストに格納された結果を受け取ることで本機構のシステムコールの発行が完了する．システムコールを処理するためのスレッドを作成することと，システムコールの発行回数を減らすことでオーバヘッドを低減させる．

3　評価

exec_thread による処理時間の推移を図 2 に示す．本実験は並列で実行する exec_thread の数を 1 から 12 個まで変化させた．7 個まではスレッドを増やすごとに処理時間が短く，並列処理の効果が確認できた．

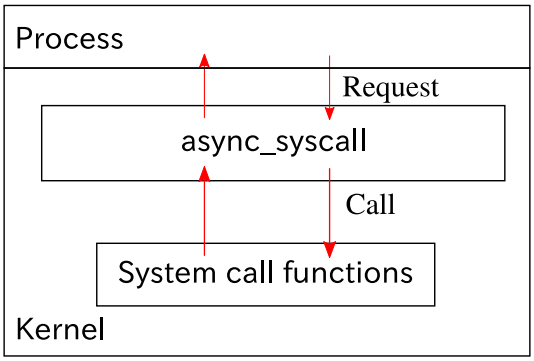

図 1　提案機構

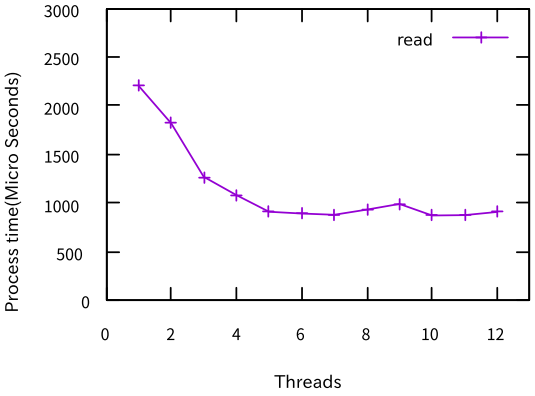

図 2　exec_thread によるシステムコール並列処理

4　おわりに

本稿では，非同期システムコール処理機構について述べた．本機構はシステムコールの処理を行うスレッドを作成することにより，システムコールの発行と処理を非同期で行うことが可能となる．また，リクエスト領域を用いてシステムコールを発行することでオーバヘッドを低減している．

参考文献

[1] Soares, L. and Stumm, M.: FlexSC: Flexible System Call Scheduling with Exception-Less System Calls., *OSDI* (Arpaci-Dusseau, R. H. and Chen, B., eds.), USENIX Association, pp. 33–46 (2010).