

論文の内容の要旨

論文題目 A Study on Language Constructs

for Implementation of Communication Protocols

(通信プロトコル実装のための言語機構に関する研究)

氏名 服部 健太

通信プロトコルを実装する場合、主として(1)プロトコルの規定するメッセージ形式に従ったパケットのデコード/エンコードと(2)プロトコルの通信手順に沿ったメッセージの送受信との2つを扱う必要がある。通信プロトコルの実装にはC言語を用いることが多いが、複雑なメッセージ形式のエンコードやデコードをCで記述するのは、困難ではないにしても退屈な作業である。また、アプリケーションが不必要にブロックすることなく、メッセージ送受信を処理するためには、多重化I/Oや非同期I/Oなど、オペレーティングシステムが提供する様々なI/O機構を適切に使用する必要がある。これらの要因により、通信プロトコルの実装には高度な知識とプログラミング技術が要求され、出来上がるプログラムも複雑なものとなる場合が多い。今日では、ウェブブラウザやメッセンジャをはじめとして、多くの興味深いソフトウェアは、何らかの通信プロトコルにしたがってネットワーク上の他のノードと通信しあうネットワークアプリケーションであり、これらを簡潔に実装するためのプログラミング言語が求められている。

本論文では、通信プロトコルの実装を支援するための言語機構として、関数呼び出しスタイルによる暗黙的な並行プロセスの起動、同期メッセージパッシングの上での透過的なI/Oチャンネル、そして、通信プロトコルのメッセージ形式にあわせた拡張正規表現とそのパターンマッチを提案する。さらに本研究では、これらの機能を備えたプログラミング言語Preccsを設計し、PreccsのソースプログラムからC言語のコードを生成するコンパイラを開発した。

通信プロトコルは本質的に並行性を備えており、その実装には並行プログラミングが適している。Preccsでは、全てのプロセスは並行に動作することが前提であり、runやspawnなどによる明示的なプロセス生成の指示は必要ない。これにより並行プロセス主体のより柔軟なプログラミングスタイルを促進する。各プロセスはPreccsのランタイムによりユーザレベルでスケジューリングされ非常に軽量であるから、プログラマはプロセスによるオーバーヘッドを気にする必要はない。透過的なI/Oチャンネルは、メッセージの送受信をPreccsのプロセス間の通信と同様に記述することを可能にする。さらに、複数のI/Oを扱うためのselectやpollといった多重化I/Oは、チャンネル間の選択実行(choice)に自然にマッピング

される。通信プロトコルのメッセージ形式は、TLV 形式とよばれる可変長のフィールドを含んだデータ構造が用いられることが多いが、通常の正規表現ではこのようなデータ構造を記述することはできない。本研究では、TLV 形式を直接的に扱えるように正規表現を拡張する。さらに拡張正規表現パターンマッチの機構によって、受信メッセージのデコードを宣言的に記述することが可能である。

本論文ではさらに、拡張正規表現のパターンマッチを高速に行なうための 2 つの手法を提案する。1 つは決定性カウンタオートマトン(DCA)によるパターンマッチである。TLV 形式のために拡張した正規表現では、可変長フィールドの記述をサポートするため、一般の決定性有限オートマトン(DFA)によるパターンマッチは不可能である。本研究では、可変長フィールドを認識するための DCA を定義し、拡張正規表現から決定性カウンタオートマトンを構成するアルゴリズムを開発した。もう 1 つはパターンマッチのスキップである。ある正規表現とマッチしたメッセージに対して、さらに狭いパターンと再度マッチングを行う場合、既にマッチ済みの事実を利用して、2 回目のマッチング処理の一部を省略することが可能になる。通信プロトコルで用いられる典型的なメッセージ形式を対象として、パターンマッチにかかる処理時間を計測した結果、これらの手法が有効であることを確認した。

本研究では、いくつかの典型的な通信プロトコルを Preccs によって実装し、記述性や性能の評価を行なった。その結果、我々の提案する機構によって、実際の通信プロトコルが簡潔に記述できることを確認できた。また、性能面に関しては、CPU 使用率において改善の余地があるものの、プロトコルの処理速度については十分実用的な性能に到達することが確認された。