レイアウト検証用ジョブ制御システム

7L－3

黒川　敦
三洋電機（株）

1．はじめに

ＬＳＩの大規模・微細化に伴い，レイアウトの検証はますます重要となっている。レイアウトの検証には，多くの処理回数，ＣＰＵ時間及びディスク容量を必要とするため，操作の簡易化，自動化そしてコンピュータ資源の有効活用が，設計効率の上で不可欠である。

当社では，市販の検証ツールの機能を十分に活かし，かつ効率良い設計を行うために，高機能ユーザインタフェイスとジョブ制御機能を備えたレイアウト検証用ジョブ制御システムを開発し，実用している。本システムの特徴は，同時に実行できるジョブ数を制限し，ジョブを交互に実行することで，ハードのレスポンスを維持しながら，制限数以上のジョブをあたかも同時に実行したかのように処理している点である。

本稿では，そのシステム概要とジョブ制御の方法について述べる。

2．システム概要

本システムの構成と機能を述べる。

2．1　構成

図1に構成を示す。

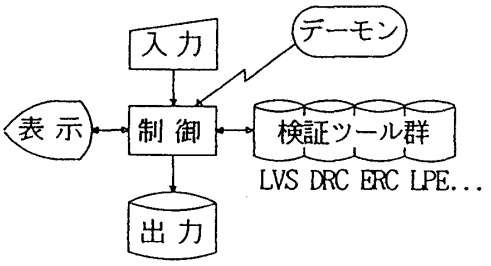

図1 システム構成

2．2　機能

本システムの機能を説明する。

（１）入力

Ｘウインドウ版とキャラクタ画面用の条件入力が可能である。ここでは，各条件（例えばファイルシンタックス）の入念なチェックを行う。

（２）出力

ヘッダに処理時間や処理の正誤の判断を記述するとともに，多くの出力ファイルから重要個所だけを抜粋し，１つのファイルにまとめる。また，利用状況の集計機能を有し，各ジョブ別，各グループ別，時間帯別などの詳細な集計も行える。

（３）表示

現在のジョブ状況をリスト表示する。内容は，ユーザ名，ジョブの種類，トランジスタ数や現在の処理ステップ数などである。ジョブの強制終了や待ちジョブの順序変更などもできる。

（４）デーモン

デーモンによりジョブ制御を定期的に実行する。デーモンは，ｃｒｏｎ［１］を用いることで実現でき，ルートの実行権で行う。

（５）制御

検証ツールのジョブ制御を行う。ジョブは，プロセスとして存在しない待ちジョブと存在する実行ジョブがある。待ちジョブには，即実行型，時間指定型及び夜間ジョブがあり，実行ジョブには，実際には実行中のものと停止中のものがある。ユーザ別制御により，特定ユーザに対して，朝までに命令の入らなかったジョブは夜間に回す機能などがある。連続ジョブのエラー判断なども行う。

3．制御方法

制御の概要と実行ジョブの制御方法を述べる。制御には３つのテーブル（表１～３）を利用する。制約（表１）は，プロセスとして存在可能な最大ジョブ数と実際に実行可能な最大実行ジョブ数とジョブの連続実行時間を制限するためのデフォルトの時分割の時間である。

A Job Control System for Layout Verifications
Atsushi KUROKAWA
SANYO Electric Co., Ltd.

表1 制御テーブル

最大ジョブ数
最大実行ジョブ数
時分割の時間

表2 待ちジョブテーブル

ユーザ名
ジョブの種類
△△を入れた時間
開始したい時間
ワークディレクトリ名
各種ファイル名
所有者
グループ
など

表3 実行ジョブテーブル

現在の時分割の時間
親PID
現在のCPU時間
前回のCPU時間
現在のステップ数

3．1　制御の概要

ジョブ制御の概要を以下に述べる。

①プロセス状況の確認

検証ツールのプロセス状況の確認と規定入力以外のプロセスの検索を行う。

②待ちジョブの実行

プロセスに存在するジョブ数が最大ジョブ数より少なく，かつ実行すべき待ちジョブ（表２）がある場合，そのジョブの所有者とグループを設定して，命令を入れる。

③実行ジョブの制御

実際に実行中のジョブ数が最大実行ジョブ数より多い場合，又は現在実行しているプロセスの連続ＣＰＵ時間が時分割の時間を越えている場合，実行ジョブを制御する。

3．2　実行ジョブの制御方法

検証ジョブは，いくつかのタイプ（ＬＶＳ，ＤＲＣ，ＥＲＣなど）と各処理が数十（例えば７０）のステップから構成される（図２）。そこで，単に時分割するのではなく，このステップ間を境にして実行の切り換えを行う。その方法を以下に示す。

①実行中のジョブ数が最大実行ジョブ数より多い場合，テーブル（表３）を参照して，現在動かしているジョブ以外を停止する。

②時分割の時間を過ぎた実行ジョブがある場合，そのステップのジョブの全親プロセスを停止し，次に停止しているジョブを復帰（継続）する。その時，前のジョブに要したＣＰＵ時間を時分割の時間として更新する。

停止と継続はシグナル（ＳＩＧＳＴＯＰ，ＳＩＧＣＯＮＴ）［１］の送信で行う。停止は親プロセスから順に行い，現在実行中のステップを完全に終了させるために，現ステップの親プロセスまで行う。継続は停止している子プロセスから順に行う。

実行ジョブ制御の概念を図３に示す。これは，最大ジョブ数が３で，最大実行ジョブ数が１の例である。本システムでは，このように時分割とステップをキーにして，ジョブを制御することがわかる。

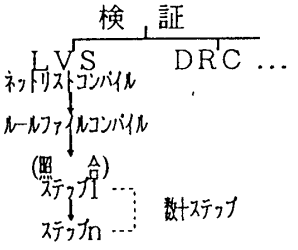

図2 ジョブ構成

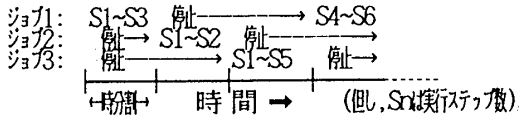

図3 実行ジョブ制御の概念

4．効果

本システムの効果を述べる。

①ユーザインタフェイスの強化により，手入力の簡素化，連続ジョブ，処理経過表示及び出力結果の見易さを実現できた。

②ジョブ制御により，無秩序なジョブ入力による処理速度の低下，ディスクやメモリオーバーを防止できた。

③実行ジョブ制御により，最大の実行ジョブ数を制限しながら，その制限数以上のジョブを処理できるので，小さいジョブの早出しが可能となった。

5．おわりに

レイアウト検証用ジョブ制御システムの機能，制御方法及び有効性を述べた。本システムでは，実行ジョブを制御することでジョブの早出しを実現しているが，各ステップ毎に分割して待ちジョブ制御だけで行う方法を現在検討中である。

最後に，日頃御指導頂く当社ＬＳＩ事業部ＣＡＤ技術部の井上部長，名野課長に感謝します。

参考文献

[1] SunOS Reference Manual, 2224p., sun microsystems, 1990