Date . . No.

スレッド

プロセス内での並行処理を実現するために取り入れられた概念，仕組み

プロセス内の仮想CPUの割り当て対象

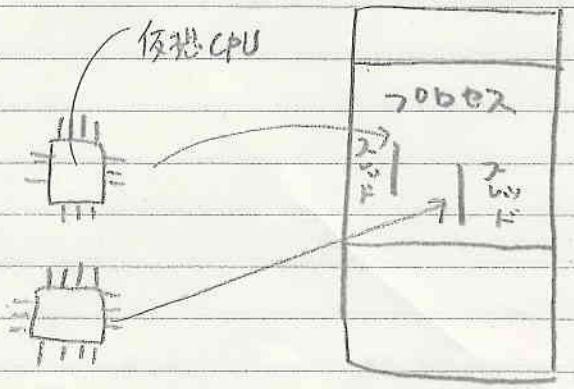
あるプロセス内に複数のスレッドが存在しているケースをマルチスレッド
と呼ぶ

マルチスレッド実行において，各スレッドはメモリ空間は共有
しているが，
専用のレジスタの状態とスタックを持つ．

スレッドへの仮想CPUの割り当ての切り替えもコンテキスト切替えという

スレッドのコンテキスト切替えはプロセス切替えと比べて小さい
メモリの切替えなどが必要ないため

このためスレッドを軽量プロセスと呼ぶ場合がある



Date . . No.

4 スケジューリング

OSはプロセスやスレッドのコンテキスト切替えを行う

どういう順序でプロセスやスレッドを実行するか決めること (CPUを割りあてるか)

↳ スケジューリング

スケジューリングを定めるソフトウェア：スケジューラー
スケジューリングで用いるアルゴリズム：スケジューリングアルゴリズム

4.1 プロセスの状態

プロセス (スレッド) のとりうる状態

① 実行状態　プロセスが実行されている状態
実際にCPU時間 (物理CPU) が割りあてられ、何らかの処理を行っている状態

② ブロック (blocked) 状態

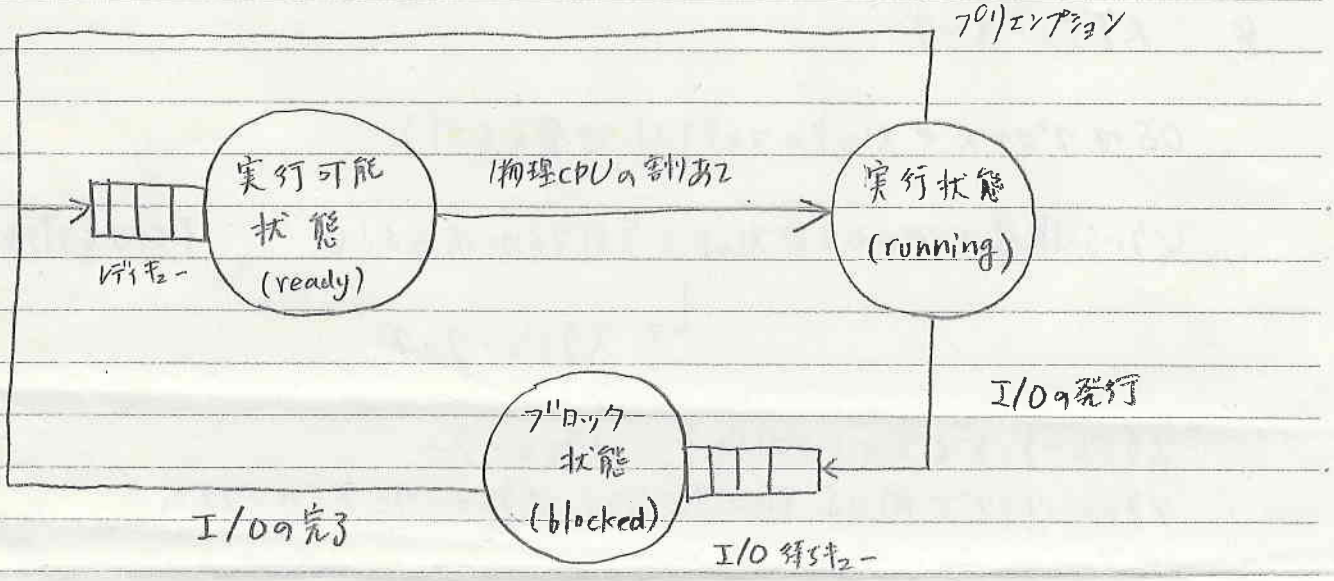
プロセスの実行が中断している状態

例えば、ディスクI/Oの完了待ちとなっていて、処理を先に進めることができない状態

③ 実行可能 (ready) 状態

プロセスの実行が中断している状態であるが、物理CPU (CPU時間) の割りあてがあれば実行可能である状態

カーネル内部では、実行可能状態にあるプロセスを待機させておくキュー (待ち行列) であるレディキュー (ready queue) や、I/O完了待ちのプロセスを待機させておくI/O待ちのキューなどがある



スケジューリングの評価基準

① ターンアラウンド時間

プロセスの実行を依頼してからプロセスの実行が完了するまでの時間

終了時刻 − 到着時刻

実行待ち時間 + 実行時間

平均ターンアラウンド時間：対象とする全てのプロセスのターンアラウンド時間の合計を当該のプロセスの数で割ったもの

		到着時刻	終了時刻	ターンアラウンド時間
(例)	プロセスA	0	5	5
	プロセスB	3	13	10
	プロセスC	6	18	12

平均ターンアラウンド時間 $\frac{5+10+12}{3} = \frac{27}{3} = 9$

Date . . No.

② スループット

単位時間内に処理されるプロセス(仕事)の数

先の例では [redacted] $\frac{3}{18} = \frac{1}{6}$

③ 応答性

対話的なプロセスの場合、キーの入力やマウスのクリックなどの対話的な操作に対して、どれだけ迅速に応答できるかということも重要

実行開始時刻 − 到着時刻 と考えられるかも

④ 公平性

特定のプロセスばかりが実行され、他のプロセスがあまりに実行されないことは避けた

常に実行可能なプロセスにいつまでもCPU時間が割り当てられない現象をスタベーション(飢餓状態)とよび、これを避ける必要がある

4.4 スケジューリングのタイミング

スケジューリングが行われるタイミングには以下の2つがある

① プロセスが自主的にCPUを解放するとき

プロセスの実行が終了したとき
ディスクI/Oなどを依頼したとき
他のプロセスにCPU時間(割り当て)を譲るyield()というシステムコールを実行した場合

↓
ready状態のプロセスの選択へ