

オペレーティングシステム実験

第7回(2009.05.28)

標準出力からのトレース(FS→kernel)最終段階

C言語のポインタ処理／配列処理

標準出力 (printf) の処理トレース 続き (3回目 / 最終段階)

□ FSからのメッセージ送信

- FSもOSの内部である。
 - (kernel, fs, mmの三つの大きな要素の一つ)
 - OSの内部でも、システムコールと同様の呼出し
 - sendrec()関数をコール: アセンブラの割り込み
-

割り込み処理は、どこで受けるか？

- Kernelにある、i8259.cのファイルで処理
 - Intel 8259とは何か
 - 割り込みハンドラ IC:
 - オペレーティングシステムは、ハードウェアに依存する部分を大量に処理する。
 - OSの上には、ハードウェアの違いを見せなくしている。
 - 「抽象化」／固有情報の隠蔽
 - <http://www.itofamily.com/ito/collections/peripherals/i8259/index.html>
 - 現在は、Pentiumプロセッサに「吸収」されている。
 - intr_init()が初期化時にコールされた後、割り込みベクトル(割り込み処理プログラムのアドレス群)を設定する。

 - この初期設定が終わると、以後、割り込みを受けるとそれぞれの「割り込み処理」プログラムが呼び出される。
-

TTY端末

- 教科書P53
 - コンピュータで、「文字」の入出力処理を行う。
 - キーボード+CRT(モニター)
 - 歴史的には、TTY+CPUの単純な構成だった。
 - それに、周辺機器として、テープ、ディスクが付加されて行った。
 - ファイル「tty.c」
 - ttyは、「常駐」して「耳を傾け続ける」
 - 何か、キーが押されなかったか？
 - 何か、表示する文字がないか？
 - そのために、無限ループの構造をとっている。
-

while(TRUE)の無限ループ

- while(TRUE){ }の無限ループ
 - OSのコードの至る所に出現する。
 - しかも、break;文がない！
 - ⇒ここから抜け出せないじゃないか!!!
 - 終了しない = OSの処理が終了するのは、
 - 「電源断」の割り込み処理が入ったとき。
 - こうした「待機プロセス」が一つずつ「消されて」いく
 - ポーリングと呼ばれる処理
 - ここから、トレースは、kernel内のdo_writeに
-

出力処理の最終ゴール

□ Kernel内のconsole.c

- cons_write()
 - put_char()
 - phys_copy()
-
- 展開されたデータを、1文字ずつ画面表示領域に物理的に転送する。
-

じっくり考えてみてください。

- 標準出力
 - `printf("Hello World!¥n");`
 - たったこれだけ処理するのに、一体何回処理の分岐、関数コールが発生したんだろうか？

 - ライブラリでは、バッファ処理
 - `printf`から`write`への変換
 - カーネルに入ってから、`cache`の処理
 - `pipe`との切り替え？
 - なぜ、`fs`から`kernel`に処理を分岐したんだろうか？

 - 全部で、30～40段階の分岐＋関数コール
-

オペレーティングシステムを読むための C言語講座

□ 構造体

- 複数の種類のデータを、一塊にして扱う。

- struct

□ 構造体ポインタ

- 「データ」を一式「カバン」に入れて持ち歩く。
- どの「カバン」を持っていくか、切り替えるような方法。

□ ポインタ演算子 `->`

- `*st.member` を `st->member`と書く。
-

文字列のポインタ (時間がない場合には省略)

- ()がついていないため、C言語の演算子の優先順位によって、実行結果を読み誤る。

*a++ = (char) b;

*++a = (char) b;

タスクテーブル

ps -ael

- と入力すると、プロセスの一覧が表示される。
 - 現在実行中のプロセス
 - ⇒ プロセス番号がマイナスのものは、何？
 - システムタスク(OSの待機タスク)
 - 画面への表示名と、比較してみよう。
-

今日の課題

- 配布プリントの空欄を埋めてください。
-

第7週:ボーナス課題:キー入力のトレース (提出期限:6月11日)

キーボードから入力があった場合、キーボードからの入力が、どうプログラムに手渡されるか。

- キーが押されると、キーボード割り込みが発生し、`kbd_hw_int()`がコールされる、というところからスタートして、OSのメッセージ配列に値が代入されるところまでを、トレースして下さい。(標準点:5点)
 - `kbd_hw_int()`が、どこで「割り込みベクトルテーブル」に設定されているかを探す。(発展課題:3点)
 - ユーザプログラムの`scanf()`が、どうシステムコールを呼び出すか、トレースする。(発展課題:標準点15点)
 - OSからのデータ受け渡し部と、ユーザライブラリでのデータ受け取り部の、つながりの部分を探す。(発展課題:7点)
-