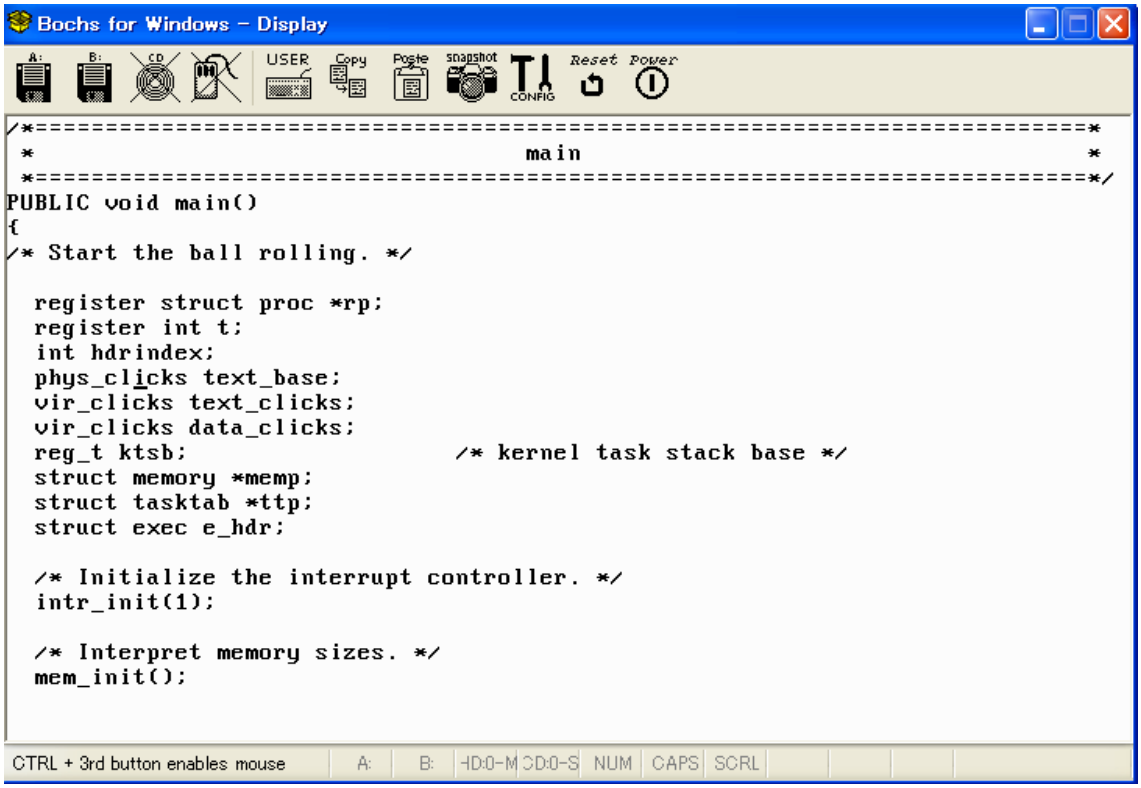
待機プロセス

以下は、/usr/src/kernel の main()プログラムの先頭部分です。



ここでは、まず、struct tasktab *ttp; で、tasktab 構造体へのポインタを定義しています。
「構造体へのポインタ」というのは、「構造体」というスーツケース（カバン）の番号をポイントするもの（指し示すもの）、何か、一塊のデータが格納されている「インスタンスのようなもの」の、どれかを指している、と考えてください。

この少し下に、

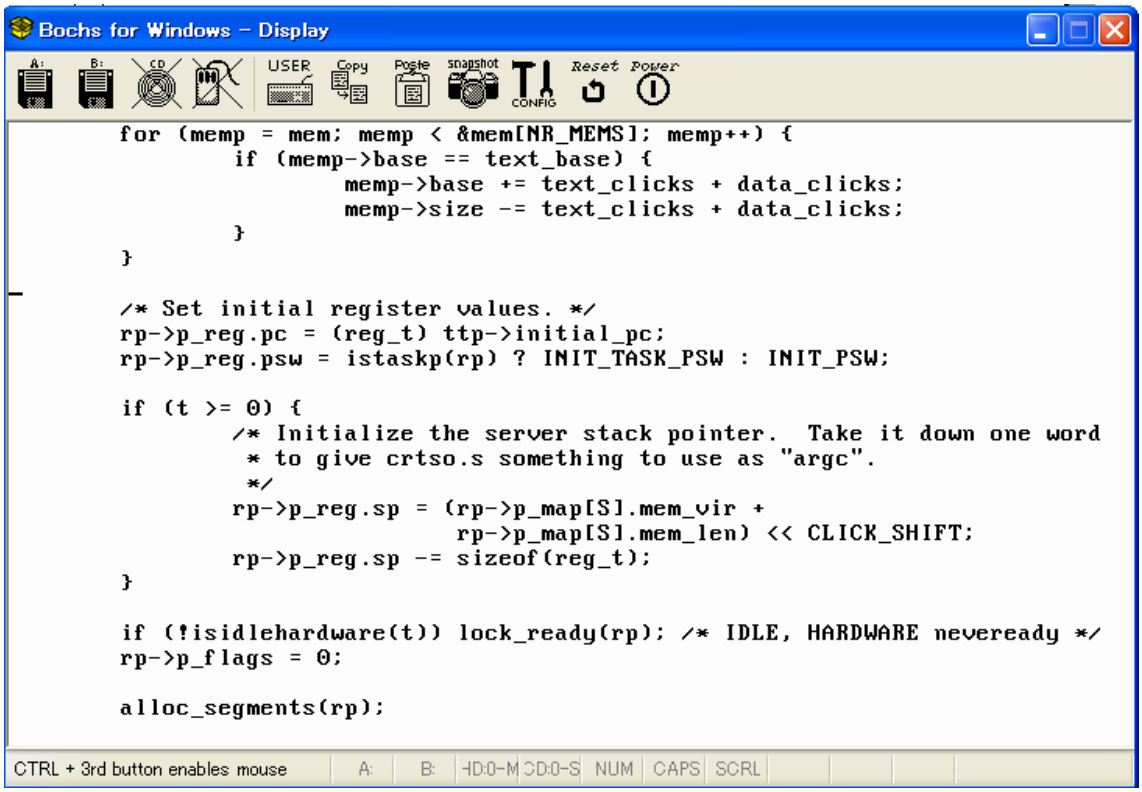
```
  /* Set up proc table entries for tasks and servers.  The stacks of the
   * kernel tasks are initialized to an array in data space.  The stacks
   * of the servers have been added to the data segment by the monitor, so
   * the stack pointer is set to the end of the data segment.  All the
   * processes are in low memory on the 8086.  On the 386 only the kernel
   * is in low memory, the rest is loaded in extended memory.
   */
```

このように書かれた部分があります。このコメントの先頭に、

Set up proc table entries for tasks and servers.

と書かれています。

さらにこの少し下、１１３行目を見ると、



rp は、proc 構造体へのポインタですが、この pc という構造体メンバに、tasktab 構造体の initial_pc というメンバ変数の値を代入しています。

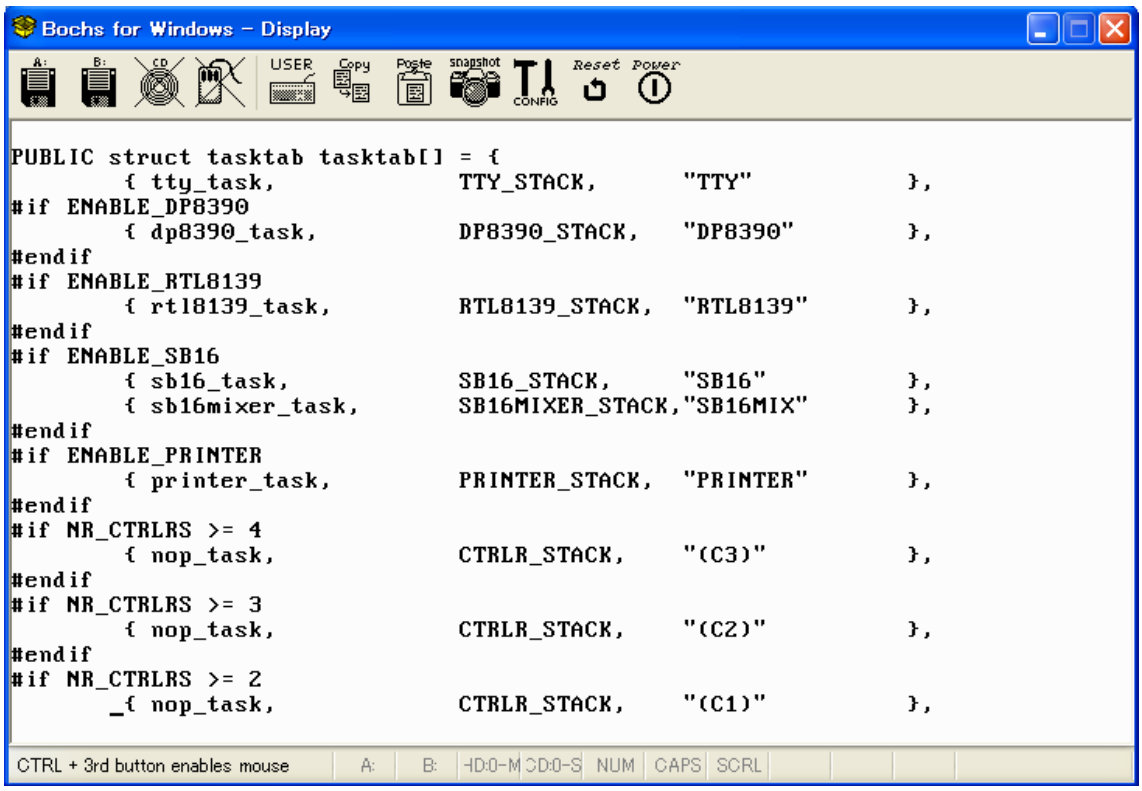
PC は、Program Counter の略で、CPU が次にどの命令を実行するか、命令語の番地を示すレジスタのことです。また、PSW は Program Status Word の略で、その命令を実行している時の CPU の状態（一つ前の計算結果で桁上がりがあったか、など）を保存している CPU のレジスタです。

rp は、プロセス群というたくさんの「カバン」を「指差す」ことのできるポインタですが、後でこの rp は一つずつが「起動」されます。
これらに値を代入する、ということは、「新たにプロセスとして起動すべきプログラム」がどこにあるかを、予め登録しているということになります。

言葉を換えると、初期化処理の中で MINIX が MINIX 自身として起動していくプロセスが、ここで準備されている、ということになります。

この、tasktab の initial_pc という変数には、どんな値が格納されているのでしょうか。

答えは table.c の 82 行目から書かれています。



type.h を見てみます。

```
struct tasktab {
  task_t *initial_pc;
  int stksize;
  char name[8];
};
```

tasktab 配列の最初の引数は initial_pc ですが、２番目は stksize であり、３番目は name(名前)です。
この名前は、どこかで確認することができます。この先は、授業での説明をよく聞いてください。