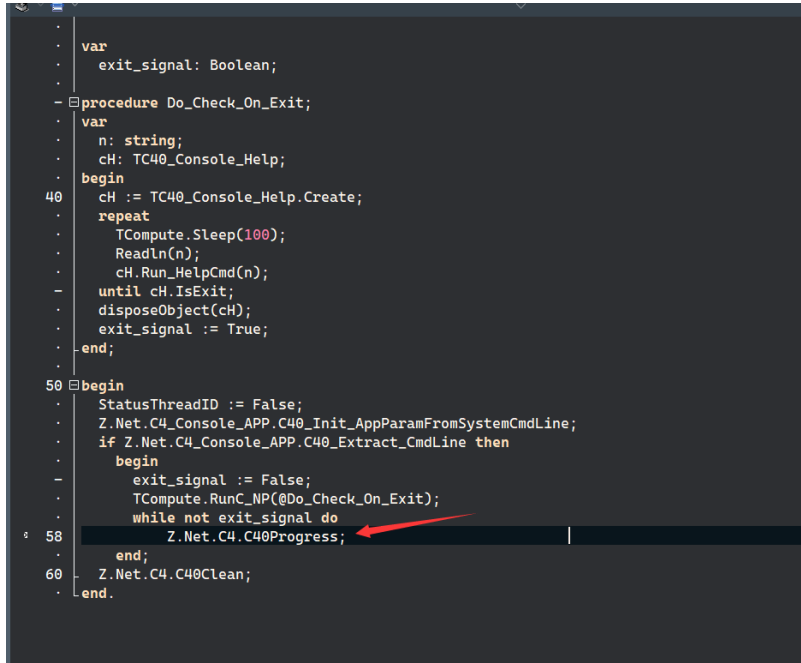


关于 C4 的主循环

C4 的整体设计路线是一致化，统一化，因此，主循环也和传统的 ZS/ZNet 不同，在 C4 中，使用主循环只需要一行。



```
var
  exit_signal: Boolean;
procedure Do_Check_On_Exit;
var
  n: string;
  cH: TC40_Console_Help;
begin
  cH := TC40_Console_Help.Create;
  repeat
    TCompute.Sleep(100);
    Readln(n);
    cH.Run_HelpCmd(n);
  until cH.IsExit;
  disposeObject(cH);
  exit_signal := True;
end;
50 begin
  StatusThreadID := False;
  Z.Net.C4_Console_APP.C40_Init_AppParamFromSystemCmdLine;
  if Z.Net.C4_Console_APP.C40_Extract_CmdLine then
  begin
    exit_signal := False;
    TCompute.RunC_NP(@Do_Check_On_Exit);
    while not exit_signal do
      Z.Net.C4.C40Progress;
    end;
  end;
  Z.Net.C4.C40Clean;
end.
```

C4 内部由各种服务器部件组成，这些部件一旦展开，多大几十种，不可以人为单独给这些服务器部件做主循环，因此 C4 的主循环采用了统一化设计，一个方法，集中解决所有问题。

C4 的主循环，不再需要绕弯子（线程同步），额外调用

- CheckThread
- CheckThreadSynchronize
- CheckSynchronize

C4 的主循环，回避了流程的时间片泄露，99%情况下，要比 ZS/ZNet 单独给服务器做循环效率更高。

如果要做出模拟阻塞的通讯机制，C4 从底层，中层，都是不支持阻塞的。只要从高级的 C4 的主循环入手，才是可取的。

by.qq600585

2022-10-15