

LabWindows/CVI

príklad s využitím výhradne ANSI C = opakovanie
znaností z predchádzajúcich predmetov aplikovaných v
prostredí LabWindows/CVI

doc. Ing. Ján Šaliga, PhD.

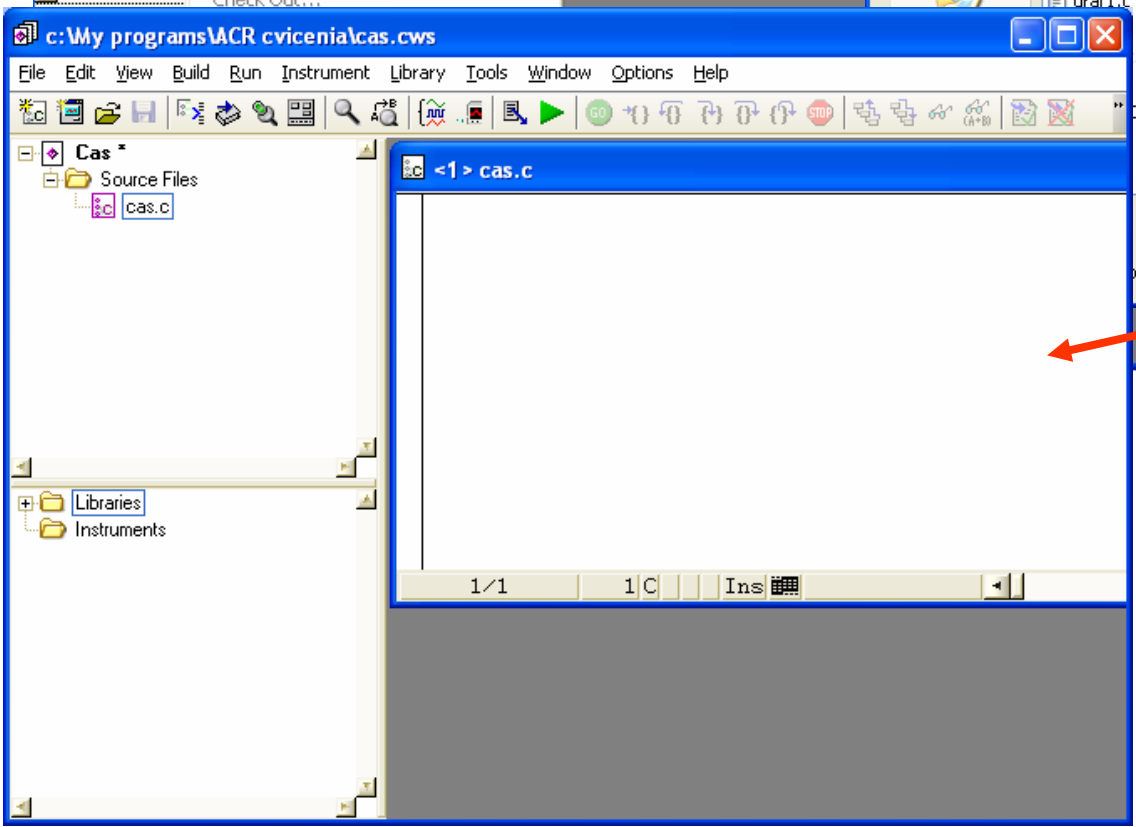
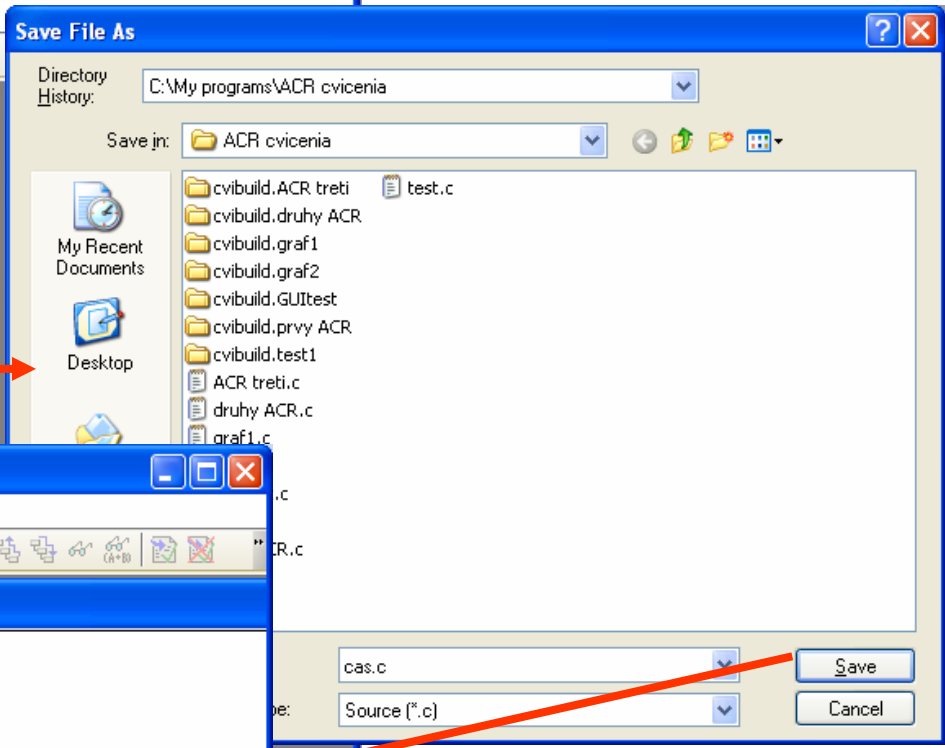
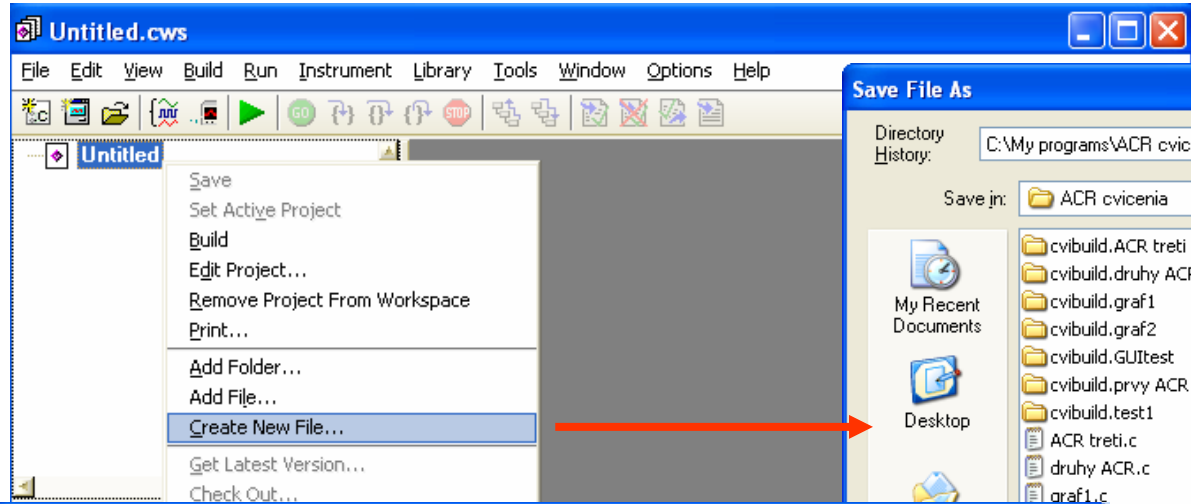
KEMT FEI TU Košice

Úloha, príprava

- Vytvorte program využívajúce iba funkcie z ANSI C (konzolový), ktorý vypíše aktuálny systémový čas počítača.

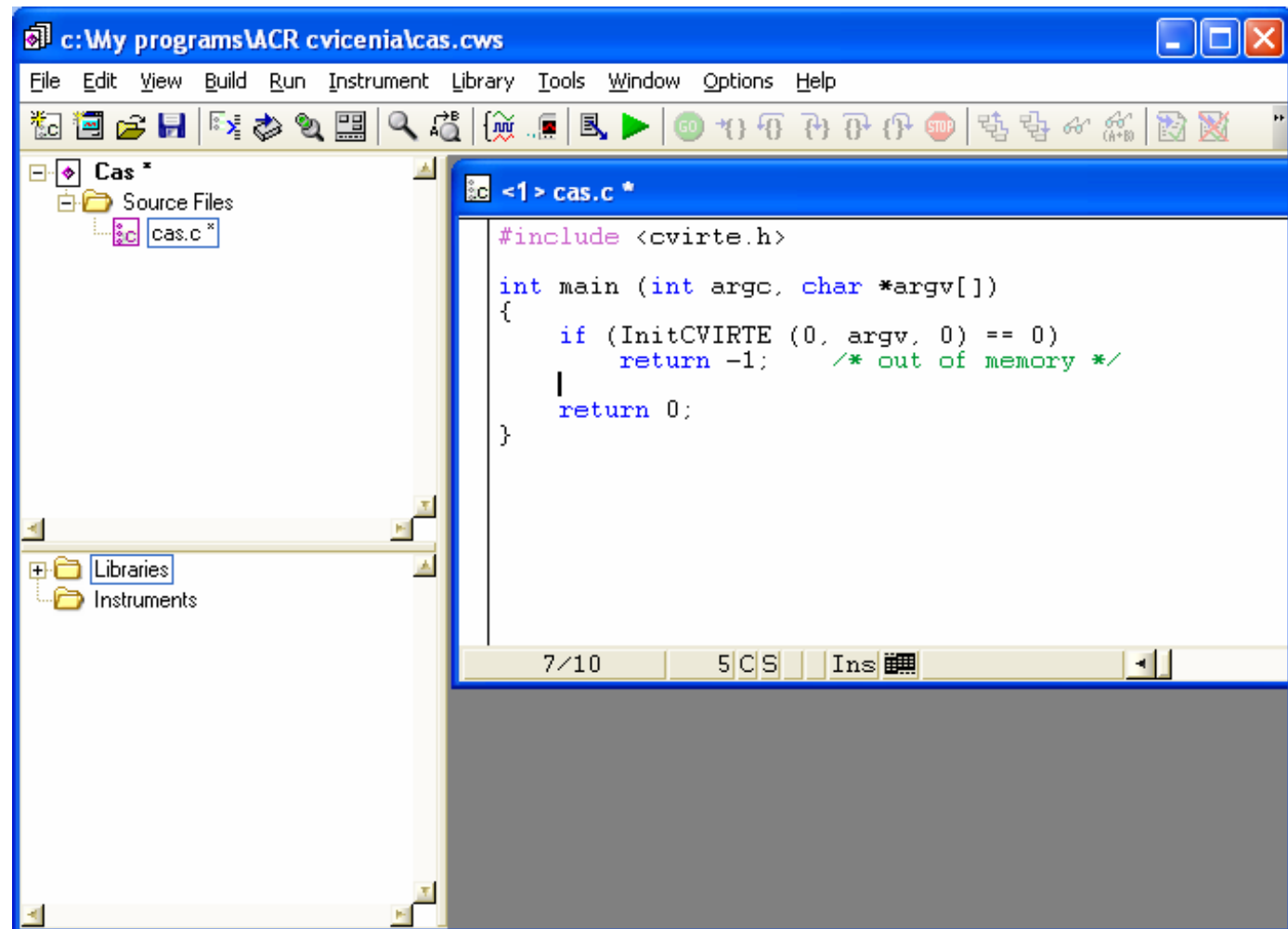
Odporúčanie: pri výbere jednotlivých príkazov z menu podľa pokynov v tomto návode, všimnite si ich ekvivalentné ikony (tlačidlá) na ovládacej lište hore pod menu a prípadne aj klávesové skratky

- Príprava:
 - Spustite LabWindows/CVI (ďalej len CVI)
 - Prostredníctvom File»New»Workspace vytvorte nový pracovný priestor, v ktorom sa automaticky vytvorí prázdny nepomenovaný (untitled) projekt
 - Kliknite naň pravým tlačidlom myši (ďalej skratka PTM) a vyberte položku Create new file
 - V dialógovom okne napíšte meno nového súboru, napr. cas.c a potvrdte – vytvorí sa súbor cas.c, otvorí sa v editačnom okne, zaradí sa do projektu, projekt sa pomenuje rovnako (až na príponu.prj – teda cas.prj)



Kostra programu

- Vytvorte kostru programu cez Edit»Insert Construct»Main
- Pozn.: funkcia InitCVIRTE je potrebná iba ak sa používa iný ako vlastný z CVI, napr. z Visual C



The screenshot shows a C programming IDE window titled 'c:\My programs\ACR cvicenia\cas.cws'. The window contains a menu bar (File, Edit, View, Build, Run, Instrument, Library, Tools, Window, Options, Help) and a toolbar with various icons. On the left, a project tree shows a folder 'Cas' containing 'Source Files' and a file 'cas.c'. Below the tree are 'Libraries' and 'Instruments' sections. The main editor area displays the following C code:

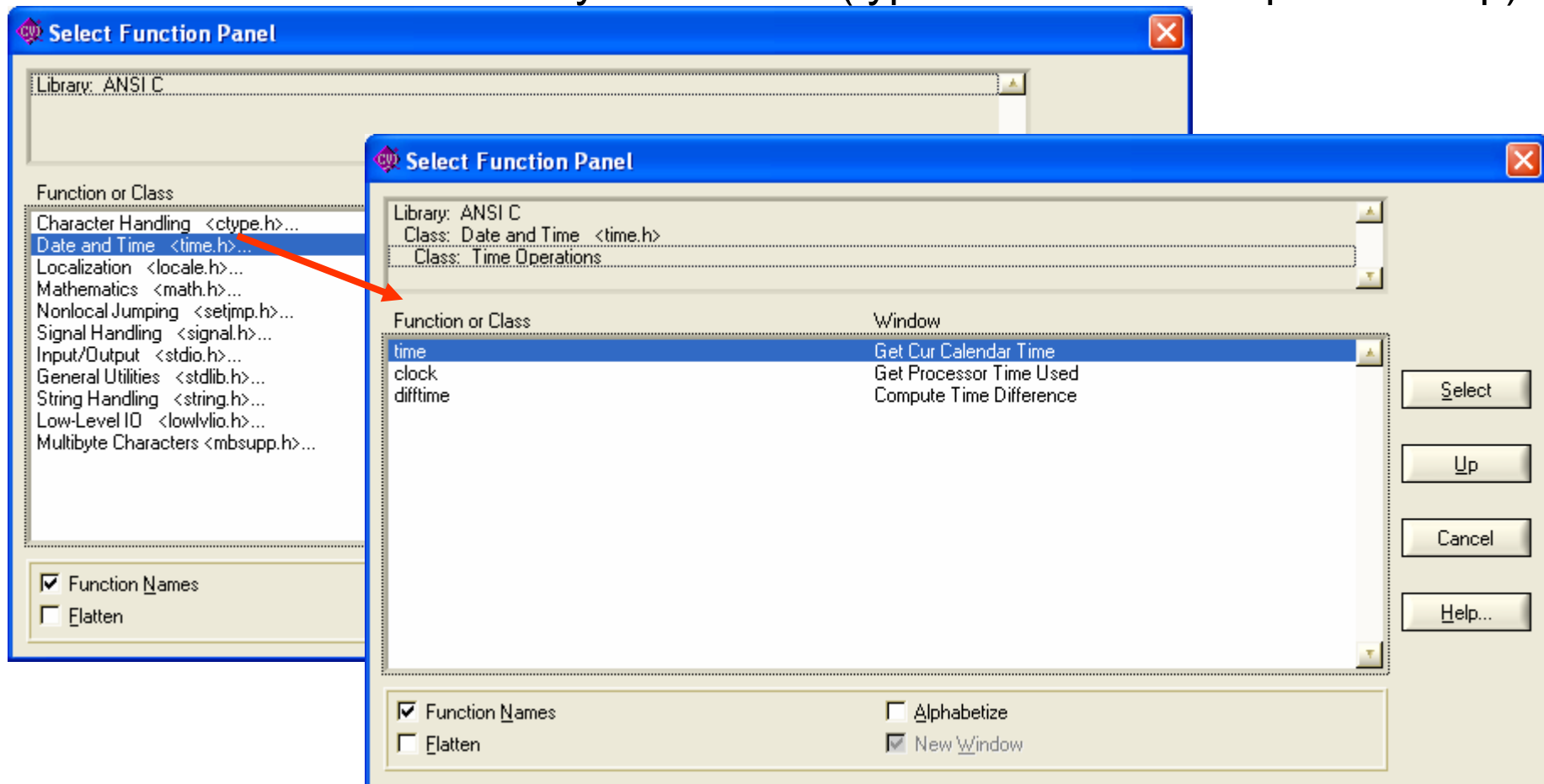
```
#include <cvirte.h>

int main (int argc, char *argv[])
{
    if (InitCVIRTE (0, argv, 0) == 0)
        return -1;    /* out of memory */
    return 0;
}
```

The status bar at the bottom indicates '7/10', '5 C S', and 'Ins'.

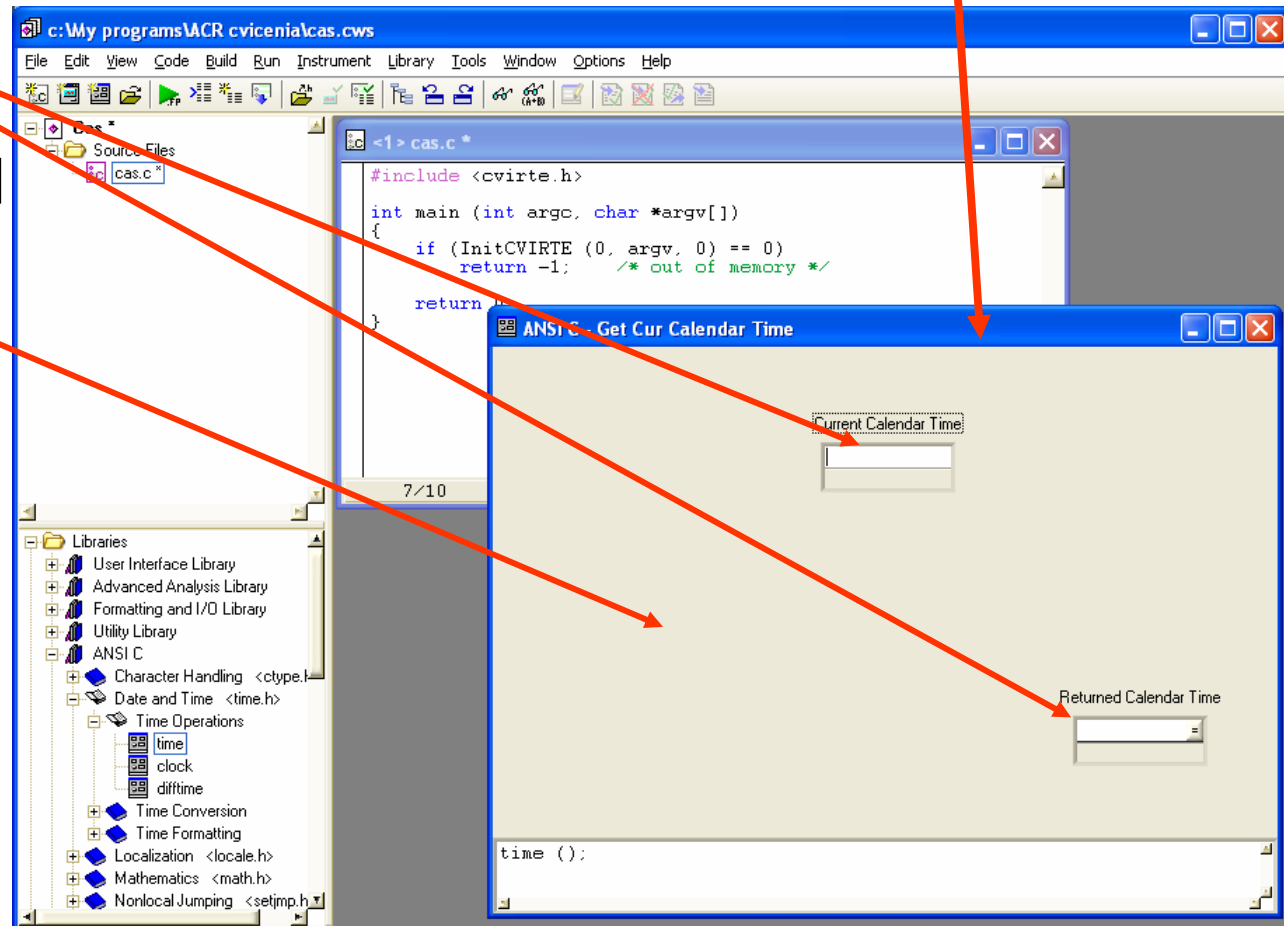
Výber a vloženie funkcie I.

- Cez Library»ANSI C sa dostanete do panelu tried funkcií ANSI C
- Vyberte triedu Date and time a následne cez Time operations sa dostanete až k samotným funkciám (typ: v každom kroku pozrite Help)



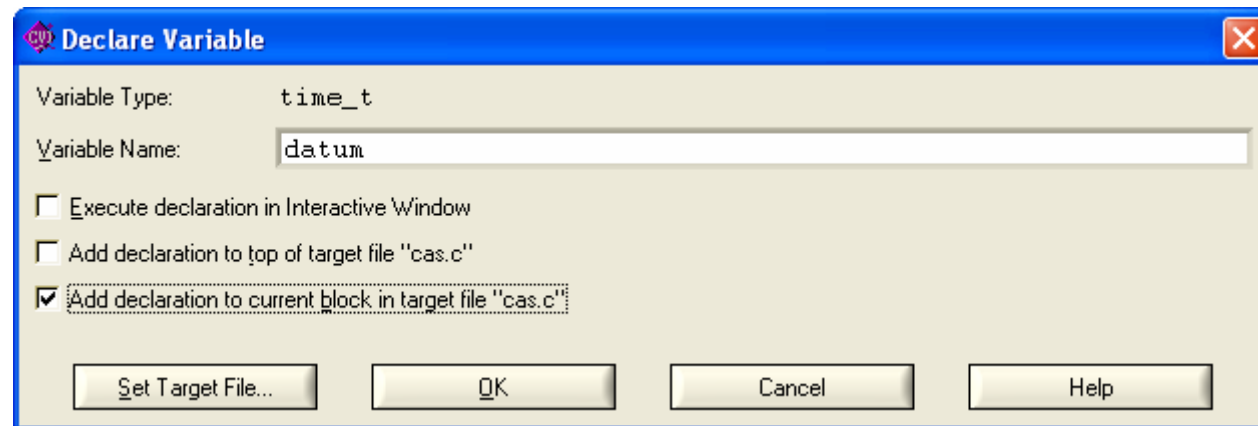
Výber a vloženie funkcie II.

- Vyberte funkciu time() a otvorí sa panel funkcie
- Kontextový Help je dostupný cez PTM k premenným funkcie a k samotnej funkcii (PTM kdekoľvek mimo premenných



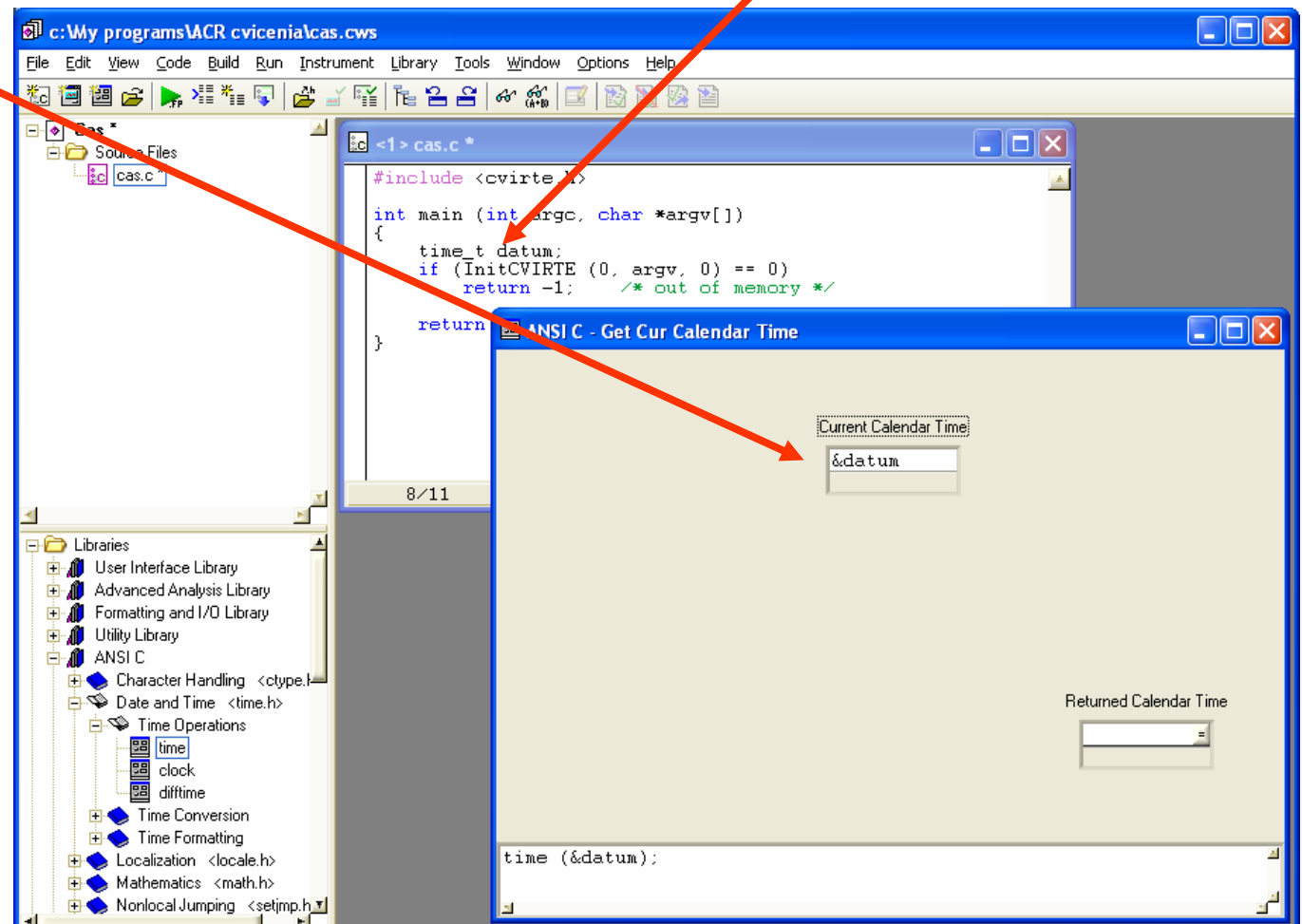
Výber a vloženie funkcie III.

- Umiestnite kurzor do okna premennej Current Calendar Date a cez Code»Declare variable (alebo cez príslušnú ikonu z menu) otvorte okno deklarácie premennej
- Napíšte zvolené meno premennej, napr. datum
- Zvoľte typ premennej lokálna globálna, napr. lokálna a potvrdíte (OK)



Výber a vloženie funkcie IV.

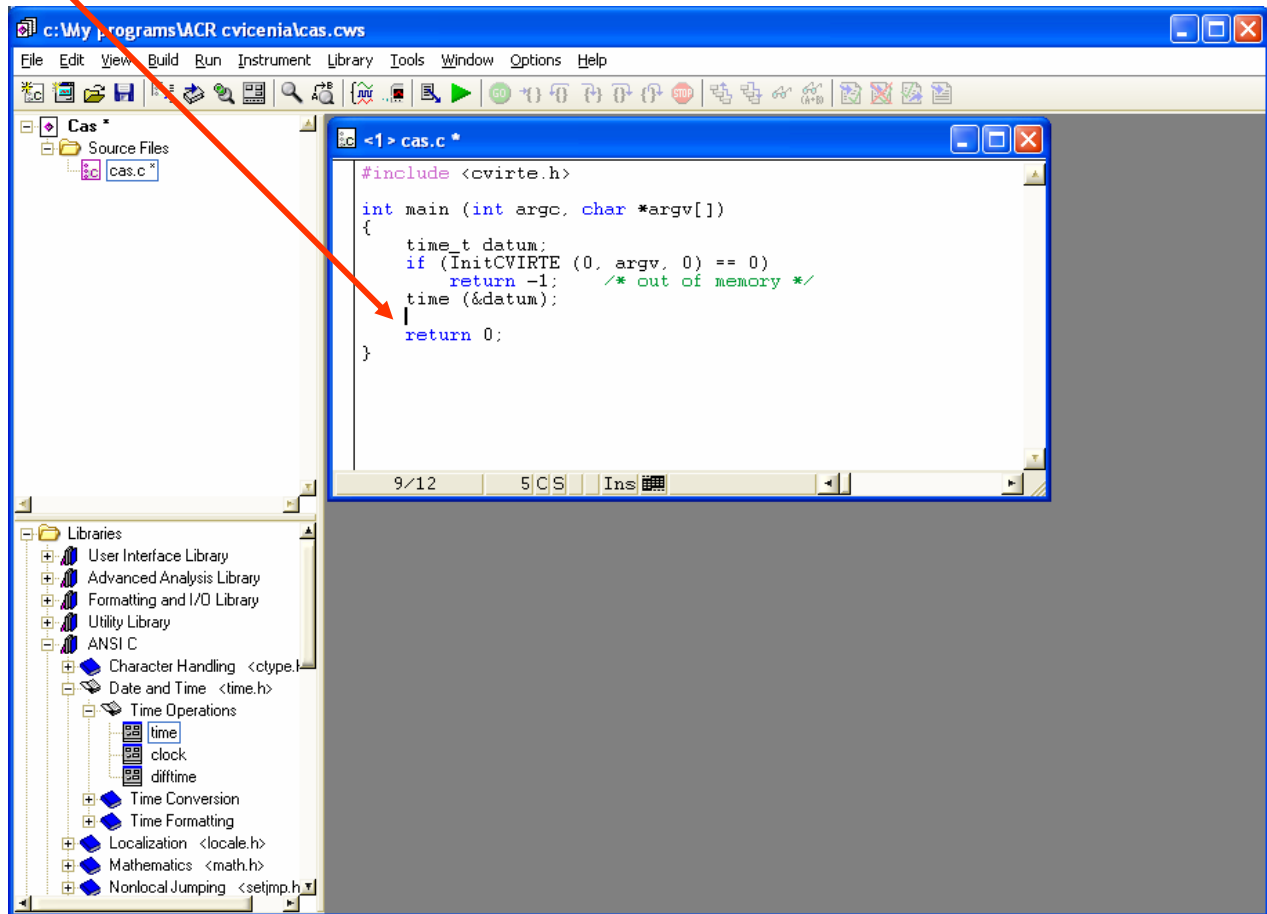
- Deklarácia sa zapíše do zdrojového súboru ako aj do políčka premennej v paneli funkcie time()



Výber a vloženie funkcie IV.

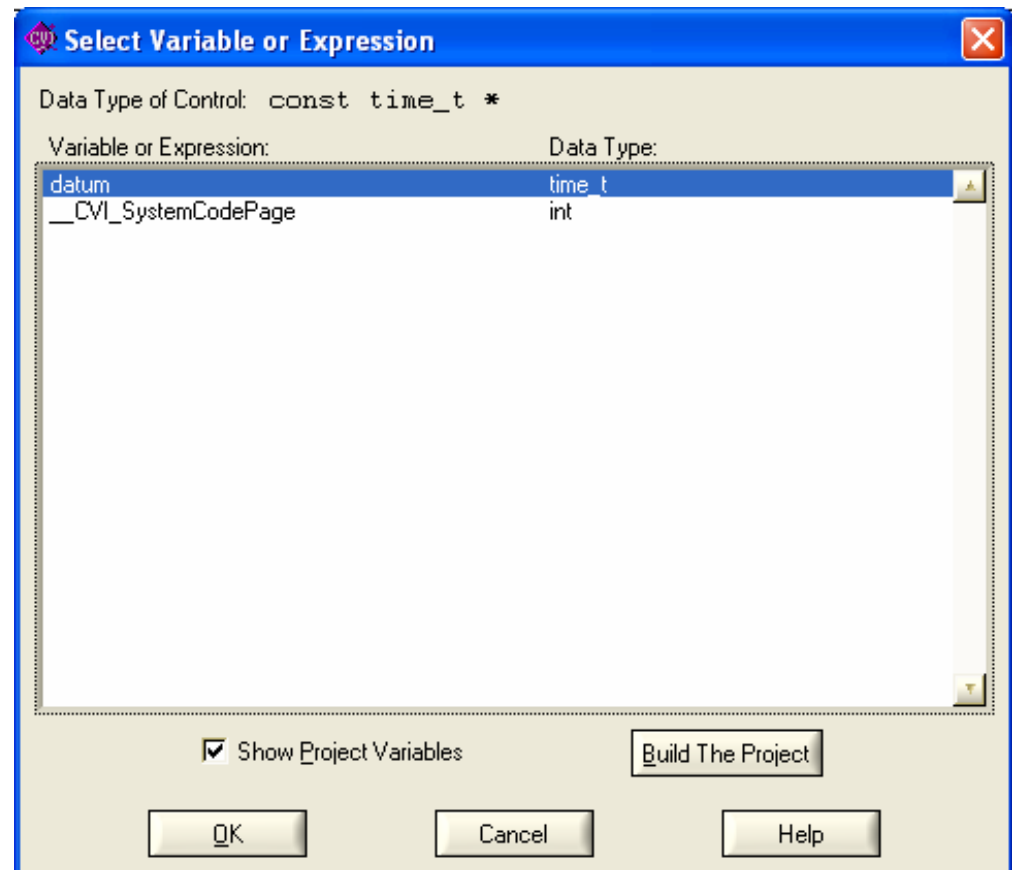
- Umiestnite kurzor z okna cas.c na miesto, kam chcete vložiť funkciu time() (nad return 0)
- Prejdite do okna Panel funkcie time()* a cez Code »Insert function call (alebo príslušnú ikonu) zapíšete funkciu na dané miesto do cas.c

* Pozri predchádzajúcu snímku



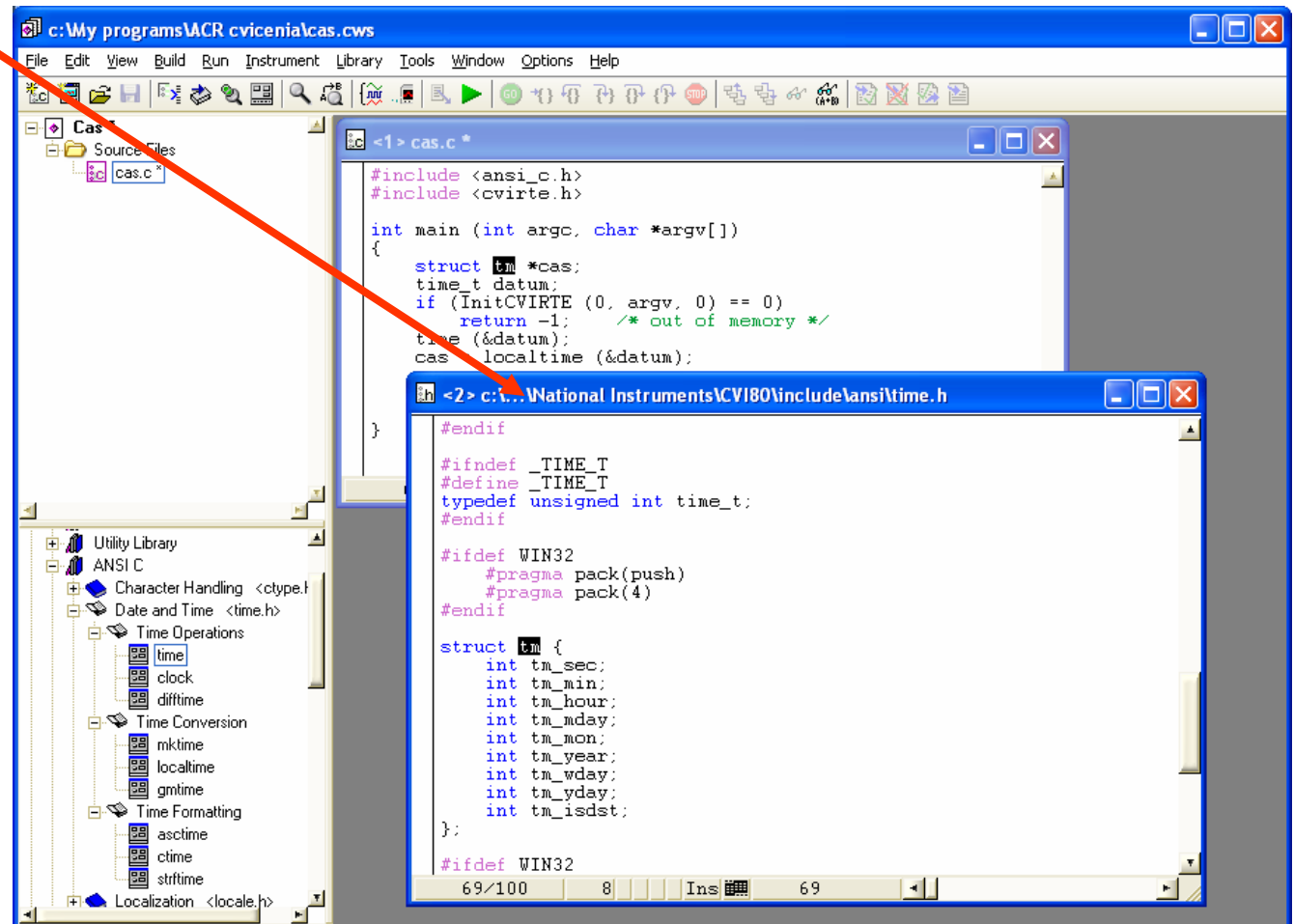
Výber a vloženie funkcie V.

- Obdobne vyberte novú konverznú funkciu localtime() z ANSI C pre konverziu hodnoty na štruktúru s rozdelenými údajmi pre sekundy, minúty, ...
- Umiestnite kurzor do okna premennej Calendar time a cez Code»Select Variable otvorte okno deklarovaných premenných. Ak nevidíte doteraz deklarované premenné, aktualizujte zoznam cez Build The project
- Vyberte príslušnú premennú, t.j. datum a potvrdte
- Deklarujte premennú Local time ako lokálnu a nazvite ju cas
- Vložte funkciu localtime() do cas.c



Výber a vloženie funkcie VI.

- Pre porozumenie štruktúry cas (struc tm) označte v deklarácii v cas.c tm a cez Edit»Go to Definition otvorte deklaračný súbor, kde je deklarácia tm
- Naštudujte zložky



The screenshot shows a development environment with two windows. The top window, titled '<1> cas.c', contains the following code:

```
#include <ansi_c.h>
#include <cvirte.h>

int main (int argc, char *argv[])
{
    struct tm *cas;
    time_t datum;
    if (InitCVIRTE (0, argv, 0) == 0)
        return -1; /* out of memory */
    time (&datum);
    cas = localtime (&datum);
}
```

The bottom window, titled '<2> c:\t...\National Instruments\CVI80\include\ansi\time.h', shows the definition of the `time_t` type and the `tm` struct:

```
#endif

#ifndef _TIME_T
#define _TIME_T
typedef unsigned int time_t;
#endif

#ifdef WIN32
#pragma pack(push)
#pragma pack(4)
#endif

struct tm {
    int tm_sec;
    int tm_min;
    int tm_hour;
    int tm_mday;
    int tm_mon;
    int tm_year;
    int tm_wday;
    int tm_yday;
    int tm_isdst;
};

#ifdef WIN32
```

An orange arrow points from the `tm` field in the `cas` struct in `cas.c` to the `struct tm` definition in `time.h`. The left sidebar shows a project tree with 'ANSI C' expanded to 'Date and Time' and 'Time Operations'.

Výber a vloženie funkcie VII.

- Vráťte sa do okna cas.c
- Vyberte funkciu výpisu printf() z knižnice ANSI C alebo priamo napíšte printf a z menu cez PTM vyberte Recal function panel
- Vyplňte položky Format String a Source(s)
(pomôcka - napíšte: "Aktualny cas je: %d:%d:%d" a ďalej cas->tm_hour, cas->tm_min, cas->tm_sec)
- Vložte funkciu do cas.c (prípadne potvrdte otázku na Replace ak ste začali podľa doporučenia písaním „printf“)

Spustenie a ladenie I.

- Vložte Breakpoint na return 0 (klik myšou do stĺpca vľavo)

The screenshot shows a development environment with a project named 'Cas' and a source file 'cas.c'. The code in 'cas.c' is as follows:

```
#include <ansi_c.h>
#include <cvirte.h>

int main (int argc, char *argv[])
{
    struct tm *cas;
    time_t datum;
    if (InitCVIRTE (0, argv, 0) == 0)
        return -1; /* out of memory */
    time (&datum);
    cas = localtime (&datum);
    printf ("Aktualny cas je: %d:%d:%d", cas->tm_hour, cas->tm_min, cas->tm_sec);

    return 0;
}
```

A red diamond breakpoint is placed on the line `return 0;`. The status bar at the bottom of the editor window shows '14/16' lines and '5' columns. Below the editor, a library browser shows various system headers and functions, with 'printf' selected.

Spustenie a ladenie II.

- Spustite program cez Run»Debug cas_dbg.exe.
- Potvrďte prípadné návrhy na doplnenie include
- Beh sa zastaví na Breakpointe
- Prepnite sa do okna Standard I/O (lišta okien vo Windows) a pozrite si výpis Vášho programu
- Dokončíte beh programu cez Run»Continue

