Subject: TASM versus TMP
Posted by Paul Förster on Wed, 01 Jan 2014 13:22:55 GMT
View Forum Message <> Reply to Message

Hi,

.... the usual macro assembler question: What is the best (macro)
assempler? ;-) No monitors, please, just macro assemblers...

I have TASM 7.4 and Turbo Macro Ass Pro 7.2. The former uses 48 blocks
while the latter uses 73 blocks. What does TMP 7.2 offer in that
additional space that TASM 7.4 does not have? One thing I found is that
file formats for load/save differ. Transferring files between the two
via export/import as SEQ works, but this is ugly. Also, TASM 7.4 comes
as an sfx loaded to $0801 which is ugly as it overwrites any loaded
BASIC program if it has to be reloaded, which is also ugly.

Also, is there any cart file available? I don't care whether I have to
start it via SYS or autostart as long as it's available after a reset.
:-) I've searched the net but didn't find anything. In fact, I found
TASM 7.4 somewhere in Russia only which is kind of exotic but of course
doesn't mean anything.

Also, happy new year everybody. :-)
--
cul8er

Paul
paul.foerster@gmx.net

Subject: Re: TASM versus TMP
Posted by &lt;address_is on Thu, 02 Jan 2014 12:20:54 GMT
View Forum Message <> Reply to Message

Paul Förster <paul.foerster@gmx.net> wrote:
> Hi,
>
> ... the usual macro assembler question: What is the best (macro)
> assempler? ;-) No monitors, please, just macro assemblers...

These days you rather use cross-development tools. The two I can recommend
are ca65 from cc65 suite and xa65. There are other as well but I personally
got used to these two. If you insist on "native" then the best (although
little known) to me was always the Input-64 Macro Assembler, very fast,
with Wordstar like editor and tons of possibilities. After using this one,
I couldn't get swayed to any TASM version of the week...

--
SD!

Subject: Re: TASM versus TMP
Posted by Harry Potter on Thu, 02 Jan 2014 15:26:30 GMT
View Forum Message <> Reply to Message

On Thursday, January 2, 2014 7:20:54 AM UTC-5, addre...@invalid.invalid wrote:
> These days you rather use cross-development tools. The two I can recommend
> are ca65 from cc65 suite and xa65. There are other as well but I personally
> got used to these two. If you insist on "native" then the best (although
> little known) to me was always the Input-64 Macro Assembler, very fast,
> with Wordstar like editor and tons of possibilities. After using this one,
> I couldn't get swayed to any TASM version of the week...
>

I agree that cross-platform tools are usually the best way to go, but, for some people, (myself included :)) there is an attraction to using native tools for compilation/assembly. If nothing else, it gives a programmer the claim of social status. For that, I recommend Power C 64/128. I forgot where I got these programs and have not yet used them, but the only quarrel I have with these is the lack of a signed shift right. BTW, I believe that these have an assembler.

Subject: Re: TASM versus TMP
Posted by Harry Potter on Thu, 02 Jan 2014 15:31:13 GMT
View Forum Message <> Reply to Message

On Thursday, January 2, 2014 10:26:30 AM UTC-5, Harry Potter wrote:
> I agree that cross-platform tools are usually the best way to go, but, for some people, (myself included :)) there is an attraction to using native tools for compilation/assembly. If nothing else, it gives a programmer the claim of social status. For that, I recommend Power C 64/128. I forgot where I got these programs and have not yet used them, but the only quarrel I have with these is the lack of a signed shift right. BTW, I believe that these have an assembler.

Found it! :)

http://www.lyonlabs.org/commodore/powerc.html

Subject: Re: TASM versus TMP
Posted by Harry Potter on Thu, 02 Jan 2014 15:44:51 GMT
View Forum Message <> Reply to Message

On Thursday, January 2, 2014 10:31:13 AM UTC-5, Harry Potter wrote:
> On Thursday, January 2, 2014 10:26:30 AM UTC-5, Harry Potter wrote:
>
>> I agree that cross-platform tools are usually the best way to go, but, for some people, (myself

included :)) there is an attraction to using native tools for compilation/assembly. If nothing else, it gives a programmer the claim of social status. For that, I recommend Power C 64/128. I forgot where I got these programs and have not yet used them, but the only quarrel I have with these is the lack of a signed shift right. BTW, I believe that these have an assembler.

>
> Found it! :)
>
> http://www.lyonlabs.org/commodore/powerc.html

Apparently, the link no longer has the 128 version. :(Can anybody help me with the 128 version? I Googled it but came up empty-handed. The site above was included in the list and didn't even contain the number and some other potential sites were inaccessible.

Subject: Re: TASM versus TMP
Posted by Harry Potter on Thu, 02 Jan 2014 16:10:46 GMT
View Forum Message <> Reply to Message

One more thing: BASSEM. It is a BASIC extension for the C64 that allows for the assembly for assembler code within BASIC. It doesn't have macro support, but I think you can emulate it through the GOSUB statement. It was issued by Compute! Gazette on April and May 1990, and its documentation spans both issues. The link follows:

http://www.jbrain.com/pub/cbm/mags/cg/

Subject: Re: TASM versus TMP
Posted by Paul Förster on Thu, 02 Jan 2014 19:25:11 GMT
View Forum Message <> Reply to Message

Hi,

.... cross development is neither an option for me nor does it feel any
original. :-P After all, I have so much programming stuff to do on
modern computers. I want the original feeling on the original machine.
Would I want to use a modern computer, then I'd use one...
--
cul8er

Paul
paul.foerster@gmx.net

Subject: Re: TASM versus TMP
Posted by Hg on Fri, 03 Jan 2014 12:58:10 GMT
View Forum Message <> Reply to Message

On 03/01/2014 00:25, Paul Förster wrote:
> Hi,
>
> ... cross development is neither an option for me nor does it feel any
> original. :-P After all, I have so much programming stuff to do on
> modern computers. I want the original feeling on the original machine.
> Would I want to use a modern computer, then I'd use one...

I see where you are coming from - though it has to be said that even
in the C64's heyday a lot of software was developed on bigger
machines even then.

--
T

Subject: Re: TASM versus TMP
Posted by &lt;address_is on Fri, 03 Jan 2014 13:54:14 GMT
View Forum Message <> Reply to Message

Hg <Hg@Hg.Hg> wrote:
> On 03/01/2014 00:25, Paul Förster wrote:
>> Hi,
>>
>> ... cross development is neither an option for me nor does it feel any
>> original. :-P After all, I have so much programming stuff to do on
>> modern computers. I want the original feeling on the original machine.
>> Would I want to use a modern computer, then I'd use one...
>
> I see where you are coming from - though it has to be said that even
> in the C64's heyday a lot of software was developed on bigger
> machines even then.

True, and I personally see nothing wrong in developing on a modern machine
and targeting the eight bitter from the past millennium. Not only I can be
much more productive using modern tools (the same rationale as back then in
the heyday) but I also find some kind of kinky pleasure in turning all
those giga/tera-things into slaves and servants of the humble breadbox ;-)
YMMV, of course. OP's too ;-)

--
SD!

Subject: Re: TASM versus TMP
Posted by Paul Förster on Fri, 03 Jan 2014 20:03:38 GMT
View Forum Message <> Reply to Message

Hi Hg, hi SD,

On 2014-01-03 13:54:14 +0000, <address_is@invalid.invalid> said:
>> I see where you are coming from - though it has to be said that even
>> in the C64's heyday a lot of software was developed on bigger
>> machines even then.
>
> True, and I personally see nothing wrong in developing on a modern machine
> and targeting the eight bitter from the past millennium. Not only I can be
> much more productive using modern tools (the same rationale as back then in
> the heyday) but I also find some kind of kinky pleasure in turning all
> those giga/tera-things into slaves and servants of the humble breadbox ;-)
> YMMV, of course. OP's too ;-)

.... I admit, I'm a little bit old-fashioned. :-P
--
cul8er

Paul
paul.foerster@gmx.net

Subject: Re: TASM versus TMP
Posted by Harry Potter on Mon, 06 Jan 2014 14:25:42 GMT

View Forum Message <> Reply to Message

On Thursday, January 2, 2014 2:25:11 PM UTC-5, Paul Förster wrote:
> Hi, ... cross development is neither an option for me nor does it feel any original. :-P After all, I have so much programming stuff to do on modern computers. I want the original feeling on the original machine. Would I want to use a modern computer, then I'd use one... -- cul8er Paul paul.foerster@gmx.net

Cross-platform *is* an option for me, but I like to program on the original.. It's a fun idea, and it would serve as as a claim of achievement. BTW, I thought to create a C compiler/assembler for the C64 and 128 using DASS and BASSEM. I don't guarantee anything, though.

Subject: Re: TASM versus TMP
Posted by Paul Förster on Mon, 06 Jan 2014 19:08:00 GMT

View Forum Message <> Reply to Message

Hi Harry,

> BTW, I thought to create a C compiler/assembler for the C64 and 128
> using DASS and BASSEM. I don't guarantee anything, though.

.... great, but please no pre-ISO notation, sometimes falsely referred to as K&R style...

good: :-)

```
int main(int argc, char *argv[])
{
    ...
}
```

bad: :-(

```
int main(argc, argv)
    int   argc;
    char  *argv[];
{
    ...
}
```

--
cul8er

Paul
paul.foerster@gmx.net

Subject: Re: TASM versus TMP

Posted by Aaron Daughtry on Mon, 06 Jan 2014 20:56:00 GMT
View Forum Message <> Reply to Message

On 2014-01-06 19:08:00 +0000, Paul Förster said:

```
>  good: :-)
>  int main(int argc, char *argv[])
> {
>     ...
> }
```

bad: :-)

```
int main (int argc, char *argv[]){
 ...
}
```

--
SD!

Subject: Re: TASM versus TMP

Posted by Paul Förster on Mon, 06 Jan 2014 21:51:43 GMT
View Forum Message <> Reply to Message

Hi SD!,

On 2014-01-06 20:56:00 +0000, SD! said:
> bad: :-)
> int main (int argc, char *argv[]){
> ...
> }

.... ok, I admit it, there's a whole huge Wikipedia article about where
to put the braces. :-) That again is a matter of taste as long as one
is consistent with oneself, isn't it?
--
cul8er

Paul
paul.foerster@gmx.net

Subject: Re: TASM versus TMP
Posted by Harry Potter on Tue, 07 Jan 2014 14:48:56 GMT
View Forum Message <> Reply to Message

On Monday, January 6, 2014 9:25:42 AM UTC-5, Harry Potter wrote:
> Cross-platform *is* an option for me, but I like to program on the original. It's a fun idea, and it would serve as as a claim of achievement. BTW, I thought to create a C compiler/assembler for the C64 and 128 using DASS and BASSEM. I don't guarantee anything, though.

Would C-like be okay? My ideas:

* tokens using the pound symbol and a letter and number
* do for now what Turbo C64 did: keep the c64/128 char set and use words for some operations
* I can keep the ANSI/ISO-style functions as is desired
* pseudo-variables: I thought of this recently: use a key-word as a variable. An example:
  sprposx[#] gets/sets the x pos. of given sprite
* instead of register, a specified zeropage location
* instead of fastcall, store parameters in abs. memory or zeropage as desired.

Don't help me out or give me any ideas yet: I still need a plan of attack. I will come here if I need help.

Subject: Re: TASM versus TMP
Posted by Payton Byrd on Tue, 07 Jan 2014 15:12:48 GMT
View Forum Message <> Reply to Message

Joseph,

Please, please, please, please, please, PLEASE read some books on compiler design before spamming this newsgroup with your ramblings about creating a compiler. You need to nail down the complete syntax, lexical analysis, and a strategy for creating assembly code for your compiled language before you should even attempt to write a compiler.

And no, I'm not volunteering to mentor you through this.

Subject: Re: TASM versus TMP
Posted by Aaron Daughtry on Tue, 07 Jan 2014 17:32:01 GMT
View Forum Message <> Reply to Message

On 2014-01-06 21:51:43 +0000, Paul Förster said:

```
>>  bad: :-)
>>  int main (int argc, char *argv[]){
>>   ...
>>  }
>
```

> ... ok, I admit it, there's a whole huge Wikipedia article about where
> to put the braces. :-) That again is a matter of taste as long as one
> is consistent with oneself, isn't it?

True. The problem starts when one has to work with someone else's code
;-) So let's make the world a better place and agree that the compiler
Harry Potter is going to write will disallow and return error once
noticing the loathsome form above ;-)

--
SD!

Subject: Re: TASM versus TMP
Posted by Paul Förster   on Tue, 07 Jan 2014 19:20:47 GMT
View Forum Message <> Reply to Message

Hi SD!,

On 2014-01-07 17:32:01 +0000, SD! said:
> True. The problem starts when one has to work with someone else's code
> ;-) So let's make the world a better place and agree that the compiler
> Harry Potter is going to write will disallow and return error once
> noticing the loathsome form above ;-)

.... I hereby would like a compiler feature that causes it to refuse to
work if certain people are in front of the keyboard. But then, that
would probably need not only a webcam but also some pretty

sophisticates asm code to check for the right person to be authorized
to use the compiler at all. ;-)
--
cul8er

Paul
paul.foerster@gmx.net

Subject: Re: TASM versus TMP
Posted by Aaron Daughtry on Tue, 07 Jan 2014 23:42:56 GMT
View Forum Message <> Reply to Message

On 2014-01-07 19:20:47 +0000, Paul Förster said:

>> True. The problem starts when one has to work with someone else's code
>> ;-) So let's make the world a better place and agree that the compiler
>> Harry Potter is going to write will disallow and return error once
>> noticing the loathsome form above ;-)
>
> ... I hereby would like a compiler feature that causes it to refuse to
> work if certain people are in front of the keyboard.

I think I know where you come from ;-)

>   But then, that would probably need not only a webcam but also some
> pretty sophisticates asm code to check for the right person to be
> authorized to use the compiler at all. ;-)

Well... 6502, and especially 6510 doesn't even have to be overclocked
to drive a Terminator, so there /are/ already some well-tested people
recognition routines in pure asm written. Imagine: somebody types

```
$ hpcc [*]
```

on his computer and the machine grabs the webcam image and posts [**]
it to the next available T-101...

A dispatcher server would need to be deployed but that's already a
trivial task these days. Any cab dispatcher software can be adapted in
no time ;-)

* - an idea for the Harry Potter's C Compiler's command-line name
** - HTTP POST
--
SD!

P. S. Just - what if he doesn't have a webcam... ??

Subject: Re: TASM versus TMP
Posted by Aaron Daughtry on Tue, 07 Jan 2014 23:45:48 GMT
View Forum Message <> Reply to Message

On 2014-01-07 15:12:48 +0000, Payton Byrd said:

> Joseph,
> Please, please, please, please, please, PLEASE read some books on
> compiler design before spamming this newsgroup with your ramblings
> about creating a compiler. You need to nail down the complete syntax,
> lexical analysis, and a strategy for creating assembly code for your
> compiled language before you should even attempt to write a compiler.
>
> And no, I'm not volunteering to mentor you through this.

Payton, Harry Potter explicitly stated that no help, let alone
mentoring, is needed! ;-)

--
SD!

Subject: Re: TASM versus TMP
Posted by Paul Förster on Thu, 09 Jan 2014 19:44:41 GMT
View Forum Message <> Reply to Message

Hi SD!,

On 2014-01-07 23:42:56 +0000, SD! said:
> I think I know where you come from ;-)

.... originally Germany, but I'm all Swiss now. :-P

> $ hpcc [*]
>
> on his computer and the machine grabs the webcam image and posts [**]
> it to the next available T-101...

.... he'd probably will have a name conflict with a certain company. But
hey, that's not a problem because this company is going down the drain
anyway, probably around the same time the T1000 arrives. 8)
--
cul8er

Paul
paul.foerster@gmx.net

Subject: Re: TASM versus TMP
Posted by Harry Potter on Fri, 10 Jan 2014 14:02:03 GMT
View Forum Message <> Reply to Message

I have a true C64C but not a C128 nor a method to send code or data to the C64C. Would it be okay to work my software on an emulator? BTW, when I said I didn't *want* help, I just meant that: I didn't *want* help for now, not that I didn't *need* help. I may come back to you for some help with the compiler later on.