

Re: How can I cause the datetime to be the name of the output file.....

## Re: How can I cause the datetime to be the name of the output file.....

---

*Source:* [http://coding.derkeiler.com/Archive/C\\_CPP/comp.lang.c/2006-02/msg00520.html](http://coding.derkeiler.com/Archive/C_CPP/comp.lang.c/2006-02/msg00520.html)

---

- *From:* Michael Mair <[Michael.Mair@xxxxxxxxxxxxxxxx](mailto:Michael.Mair@xxxxxxxxxxxxxxxx)>
  - *Date:* Sun, 05 Feb 2006 10:02:03 +0100
- 

Keith Thompson wrote:

"Merlin" <[mrb7494@xxxxxxxx](mailto:mrb7494@xxxxxxxx)> writes:

I realize that I'm new to the group, but I'm hoping that someone might be able to help me out. What I'm doing is using the GNU 7zip command line utility to make a backup on my desktop. Essentially, I'm running an executable that will create a folder, and drop a zipped up copy of my data to that folder. There doesn't seem to be a way to get 7zip to automatically create a file where the filename.zip would be a date/time stamp (something like 020206143260.zip).

What I'd like to be able to do is allow 7.zip to create the file with some arbitrary name, and then use the rename() function and the time.h library to somehow grab the current date/time from the computer and rename the file with the filename being the date/time stamp.

Would time.h be the right library to use? If so, does anyone have any suggestions as to how a solution might be implemented? Any thoughts would be greatly appreciated.

```
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>
```

```
main()
{
```

```
  chdir("C:\\Documents and Settings\\All Users\\Desktop\\");
  mkdir("Backup");
  chdir("C:\\Program Files\\Backup Utility\\");
  system("7za.exe a -tzip \"C:\\Documents and Settings\\All
  Users\\Desktop\\Backup\\Backup.zip\" \"C:\\Program
  Files\\Data\\*.\\*\\*");
  chdir("C:\\");
```

Re: How can I cause the datetime to be the name of the output file.....

```
}
```

A lot of this (<unistd.h>, 7zip, anything having to do with directories) isn't covered by standard C, and so is off-topic, but the core of your question is topical.

Use "int main(void)" rather than "main()".

Check for, and handle, errors on your function calls.

Add a "return 0;" at the end of the program. You might also do a "return EXIT\_FAILURE;" or "exit(EXIT\_FAILURE);" if you detect an error (this requires <stdlib.h>).

Yes, <time.h> is the header (not library) that you want to use to deal with timestamps. Call time() to get a raw timestamp, type time\_t, then use either gmtime() or localtime() to convert the time\_t t