WARNING: The due date on this programming assignment is firm. No late or resubmitted assignments will be accepted after that date.

Programming Assignment #5 **Due: 5/6, 5pm**

1. [100 points] Write a program called `checklinks` that will test all hyperlinks in a web page or an HTML file for validity. This is a handy program if you develop or maintain web pages.

   The `man` page follows

   Notes:

   - Create a regular expression to match URLs found in links that follow this pattern:
     - a `'<'`
     - 0 or more instances of any character except `'>'`
     - the string `"href="`
     - † an optional `'"'`
     - an RE group denoting the URL to be extracted:
       * the string `"http"`
       * an optional `'s'`
       * a `':'`
       * 0 or more instances of any character except `'"'`, `'?'`, or `'#'`.
     - 0 or more instances of any character except `'"'`
     - iff a `'"'` was found at †, a matching `'"'`
     - 0 or more instances of any character except `'>'`
     - a `'>'`

     Be sure to follow C language string escape conventions, especially for `'"'`, and allow multiple matches on the same line.

     You may want to have the instructor verify your RE.

   - If the "`-p`" flag is passed, `checklinks` works in parallel: For every URL found, the parent calls *fork(2)* followed by an `exec*(3)` (pick one) of *wget(1)* in the child. Once *all* of the children are created, the parent then waits (i.e., *wait(2)*) for them to complete and then collects their status. This allows each *wget(1)* instance to execute independently of the others in parallel.

     As it happens, in parallel mode `checklinks` usually creates a lot of zombie processes. See why? (There's nothing wrong with this.) You can watch them with and *top(1)*.

     In a `README.txt` file, document the large time savings (with identical results) you get operating in parallel mode. **Efficient, parallel operation is the most important part of this programming assignment and is worth 15 points.**

- To download the contents of a URL *url* to standard output, use

  ```
  wget --no-cache --delete-after -q -O - url
  ```

  You can try this from the command line. You may want to put single quotes around *url*.

- To check for the presence of a URL *url*, use

  ```
  wget --spider -q --delete-after -T10 -t1 url
  ```

  and check the return status. For efficiency (but not complete accuracy), the "`-T10`" sets the timeout to 10 seconds and the "`-t1`" limits `wget` to a single try. You can try these from the command line as well.

- If the URL ends with a "`/`", remove it.

- Tell *regcomp(3)* to ignore case and use "extended regular expression" syntax.

- If you run this program on `elec`, the system limits you to 100 processes, including children, so through no fault of your own *fork(2)* might fail on a page with a lot of links. If you run this on a `cslab` (`https://remote.tricities.wsu.edu`) virtual machine, the limit is much larger (> 30,000!).

NAME

`checklinks` - check every link on a web page or in an HTML file for validity

SYNOPSIS

`checklinks [`*`option`*`]*` *`urlOrFilename`*

DESCRIPTION

`checklinks` retrieves the contents of a web page or reads a file and scans the result for URLs (hyperlinks). Each hyperlink is then tested for existence. Finally, ***`checklinks`*** prints out all of the links, sorted uniquely, with each URL prefaced by either "`okay`" if it was accessible or "`error`" if it was not.

Options are:

-f treat the *`urlOrFilename`* argument as a (local) file name. (default: Treat it as a URL.)

-h print a help message and exit

-p run in parallel

ERRORS

`checklinks` notes these errors by writing an appropriate message to standard error and then exiting with an error status indicating failure:

- if it can't retrieve a URL or open a file
- if it can't execute *wget(1)*
- if any system call fails

EXAMPLE

Here is the result of running `checklinks` on the course web page (with a long line wrapped for this man page):

```
okay   http://www.tricity.wsu.edu/disability
error  http://www.tricity.wsu.edu/this_link_does_not_exist
okay   http://www.tricity.wsu.edu/~bobl
okay   https://communitystandards.wsu.edu/policies-and-reporting/
         academic-integrity-policy
okay   https://oem.wsu.edu/about-us
okay   https://oem.wsu.edu/emergency-procedures/active-shooter
okay   https://provost.wsu.edu/classroom-safety
okay   https://remote.tricities.wsu.edu
error  https://tricity.mywconline.com
okay   https://wsu.edu/covid-19
okay   https://wsu.zoom.us/j/5635500668
error  https://www.tricity.wsu.edu/cs/cslab.html
okay   https://www.youtube.com/watch
```