

John McPhee and KEdit

1. Introduction

John McPhee has written about the way he's written all his articles and books.

He collected material and wrote notes in 4x6 notepads, dozens of them. He used to type up all his notes on a typewriter, in the process running through them once again in his mind, and then make a copy of them. He marked up the copy with codes and notes based on how he thought the structure should go.

Then he cut up the copy into slivers of paper. He then laid out the slivers and organised them into sections, putting each collection into an envelope. Then he wrote again rearranging and using the slivers to place his ideas.

In 1984, a colleague of his, Howard J. Strauss, customised an editor called KEdit to do this task.

2. Structur and Alpha

McPhee's customised editor has two main parts. The first thing he does is write all his notes in a text file, the same way he did when he used a typewriter. Then he goes through and codes the notes.

Structur goes through that file and explodes the notes, creating as many files as there are codes. McPhee can then go and edit each of those files.

The other part of this is alpha, which combines the exploded files back together again to create a single document.

3. Doing this in the GNU/Linux environment.

This is a very nice idea if you like writing in text files in a line oriented way, like I do. But I'm not going to buy KEdit and neither, I suspect, are you.

So I thought that I'd put together a couple of scripts that did the job, and here they are if they are of any use to anyone else.

4. Structur

This version of structur is a bash script that calls an awk script. The way to use it is to use

codes that start with a dot. For example, with a `.code` on a line by itself to designate the next paragraph.

Here's the code for the bash script.

```
#!/bin/bash
# Script to call structur

if [ "$1" == "" ]; then
    echo "Usage: structur codedfile"
    exit 0
fi

look . $1 | awk '!x[$0]++' > ".alpha."$1
structur.awk $1

# You don't want to edit
# the main thing and the nodes
# at the same time

stamp=`date +%Y%m%d%H%M%S`
mv $1 ".bak$stamp.$1

# But maybe keep it there
# just to make building the node easier

touch $1
```

And this is the awk script that it calls.

```
#!/usr/bin/awk -f
# Structur in Awk

BEGIN {
    code = ".uncoded"
}

{
    if ( $1 ~ /^./ ) {
        code = $1
        next
    }

    outfile = FILENAME code
    print $0 >> outfile
}
```

5. Alpha

Alpha pulls all the nodes back together again into the main file and deletes the nodes themselves.

```
#!/bin/bash
# alpha puts everything back together

if [ "$1" == "" ]; then
    echo "Usage: alpha root"
    echo "example document.txt"
    exit 0
fi

stamp=`date +%Y%m%d%H%M%S`
mv $1 ".$bak$stamp.$1 # belt and braces

# Node order is in the file
# alpha.root
nodeOrder=".alpha."$1

# Read files in the order
# they are in the node file
# and put it back together
while read line; do
    f=$1$line
    echo $f
    echo $line >> $1
    cat $f >> $1
    rm $f
done < $nodeOrder

# Remove node order
# as the next order
# should be run from structur again
rm $nodeOrder
```

6. Usage

The basic idea is to start with a text file, code it and then use `structur` and `alpha` to take your notes, break them into smaller files and then recombine them.

The one thing to note is that the codes are created and put back in the order in which they are in the file that's broken up using `structur`.

Use at your own risk, but I hope it helps.

About the author

Karthik Suresh is a Management Consultant who helps customers with energy, utility, sustainability, research, innovation and knowledge management projects. His experience includes working with large and small organisations to select

and implement strategic decision systems, improve and develop management capability and deploy risk management, IT, communications and information systems projects.

Please feel free to connect on LinkedIn or read more at www.karthik-suresh.com