Charles Zhang

ID: 305 413 659

Start Time: 10:13

End Time: 11:53

Signature: 

Scanned with CamScanner

Charles Zhang

1a) # G->good words, B->bad words, F->file, a=$0, b=$1, f=$2

```
#!/bin/bash
sort linux.words > sorted.words
good=`cat tr -cs 'A-Za-z' '[\n*]' | sort -u | comm -23 - sorted.words`
bad=`cat $2`
goodCount:0
badCount=0
str=""
if [-a $2 || $0 -lt 0 || $1 -lt 0] #edge cases for parans
then
    return 0  #return some fail condition
fi

while [$goodCount -lt $0] #good words
do
    temp=`shuf -n 1 $good`  #get result of random good word into temp var
    str="$str$temp"  # concatenate w/ current output
    str="$str "  # add a space
    good=$((good+1))  #increment counter
done
while [$badCount -lt $0] # bad words
do
    temp=`shuf -n 1 $bad`  # get random bad word
    str=``$str$temp"  # concatenate
    str="$str "  # add space
    bad=$((bad+1))  #increment counter
done
str=`shuf -e $str`  #shuffle string for random order?
echo $str  #output result
return 0
```

Scanned with CamScanner

1b) #N = num cases, G = num good, P = num bad
$0 $1 $2

```
#!/bin/bash
  i=0
while [$i -lt $0]
   touch temp #create file to store words
   genspelldata $0 $1 $2 | myspell > temp # get all bad words into temp
    if [wc -l temp != $2] #check if badwordsfound is equal to actual num
    then
       return 0
    fi
   i=$((i+1))
done
return 1
```

Scanned with CamScanner

1c)

- Hard code known test cases into genspelldata
- Add an extra parameter to check if reg-testing is occurring

7

Scanned with CamScanner

2)

```
#! usr/bin/env python3

import argparse, random, string, sys

class gospell:
    def __init__(self, filename):
        f = open(filename, 'r')
        self.lines = f.readlines()
        f.close()



def main():
    parser = ArgumentParser()
    parser.add_argument('filename', nargs='?', type = argpark.FileType('r'))
    parser.add_argument('--good', '-G', nargs=1)
    parser.add_argument('--bad', '-B, nargs=1)
    args = parser.parse_args(sys.argv[1:])
```

Scanned with CamScanner

3a)

M-x delete-horizontal-space and M-\ both call the function delete-horizontal space

Scanned with CamScanner

3b)

```
(defun delete-horizontal-space (&optional backward-only &optional forward-only)
  (interactive "*P")
  (let ((orig-pos (point)))
    (delete-region
      (if backward-only
          orig-pos
          (progn
            (skip-chars-forward " \t")
            (constrain-to-field nil orig-pos t)))
      (if forward-only
          (progn
            (skip-chars-backward " \t")
            (constrain-to-field nil orig-pos t))
          orig-pos)
      (progn
        (skip-chars-backward " \t")
        (constrain-to-field nil orig-pos)))))
```

Scanned with CamScanner

3c) bruh

```
(defun delete-horizontal-space (&optional backward-only)
  (interactive "*P")
  (let ((orig-pos (point)))
    (delete-region
      (if backward-only
          orig-pos
        (progn
          (skip-chars-forward " \t")))
      (progn
        (skip-chars-backward " \t")))))
```

Scanned with CamScanner

4)

The client-server model utilizes a system where one piece of the system is the server and the other pieces are the clients. Clients will typically send a request submitted by the user to the server and receive a response back.

wget is this request that we have the client send to the server.

Some pros of wget include its ability to run in the background of other processes, allowing us to simply send a request from the client and let the server/wget handle the work when transferring data. In addition, wget communicates with the server to support resetting, where downloads to the client will retry until the whole file is retrieved.

The danger of using wget is a potential risk of using up all disk space/ other resources, as the client and server are simply transferring data, therefore other communication between them relating to this process.

Scanned with CamScanner

5a)

This function renders a single <button> element of class "square", and is passed the click behavior of props.onClick. This button also contains the value of the prop passed to Square. These allow what ever calls Square() to pass data to it through the props variable.

This function makes use of the button element, along with class and click idenifiers. It also makes use of components in {props.onClick} and {props.value}.

5b)

```
class Square extends React.Component {
  render(){
    return (
      <button
        className="square"
        onClick={() => this.props.onClick()}
      >
        {this.props.value}
      </button>
  )
}
```

Scanned with CamScanner

6)

During development, dependencies can be used to give you less work by giving you access to code that other people have written already. On the other hand, parallelism requires you to actively think about making your code more parallel-friendly to achieve maximum benefit.

During building, dependencies must all be resolved in order for the code to build successfully. In this way, dependencies complicate the build process. Parallelism can be used to speed up the build process through parallel builders, allowing for a decrease in overall build time.

Dependencies will work to resolve themselves using existing data on your machine during installation, performing an efficient install. On the other hand, parallelism focuses on installing as quickly as possible, valuing speed over efficiency.

Development dependency -> accessing ReactDOM and the renderCS function

Build/Installation dependency -> Node.js was required to build our chocos-lapilli project

Scanned with CamScanner