Towards a Smarter Bash

HCI

Human

Computer

Barbara Blackburn

$$200 \text{ wpm} = 200 * 5 * 8 / 60$$

$$= 133 \text{ bits/s}$$

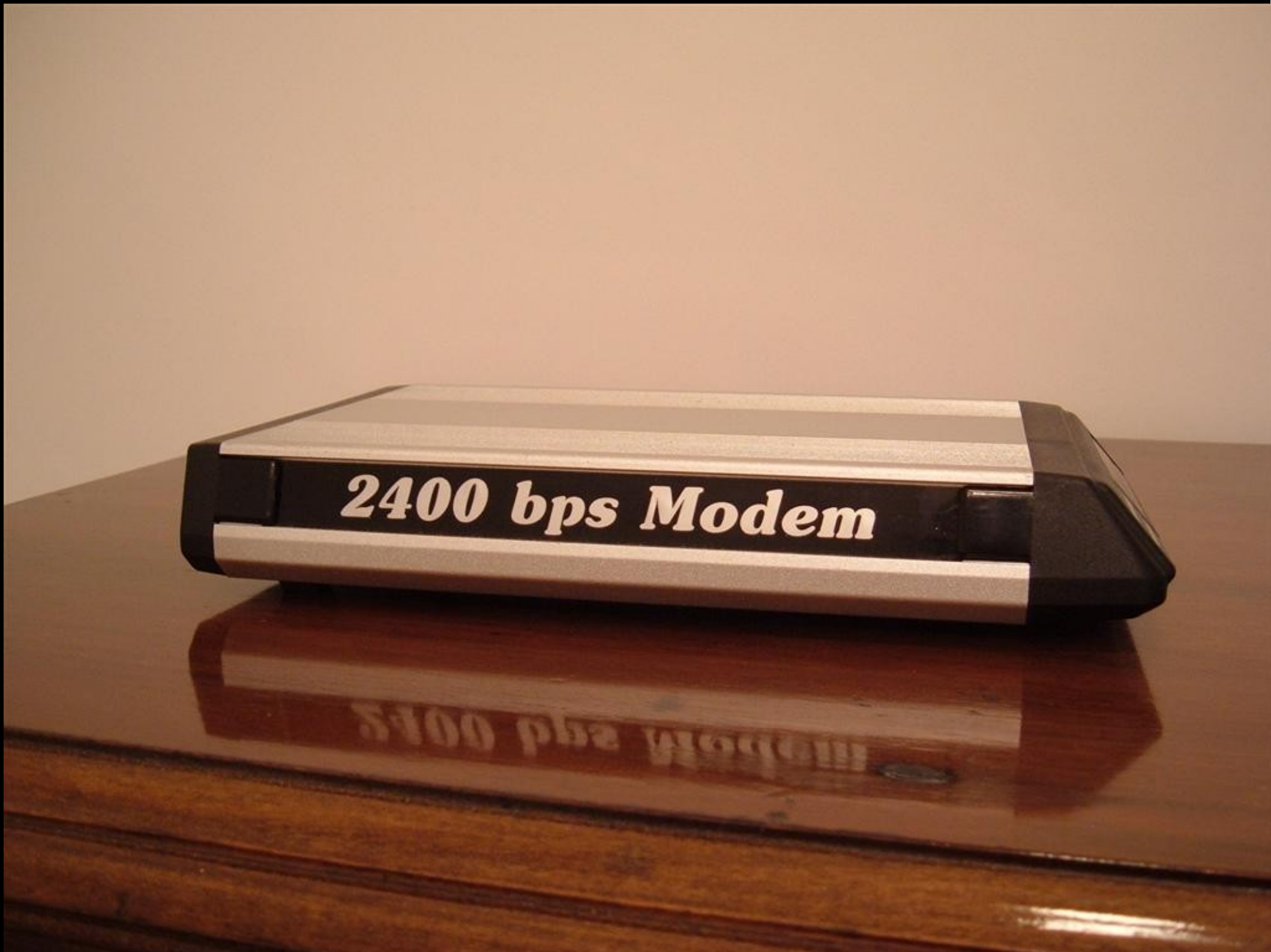

AT&T
CAT
READY
POWER
NOVATION
CAT

Howard Stephen Berg

25,000 wpm

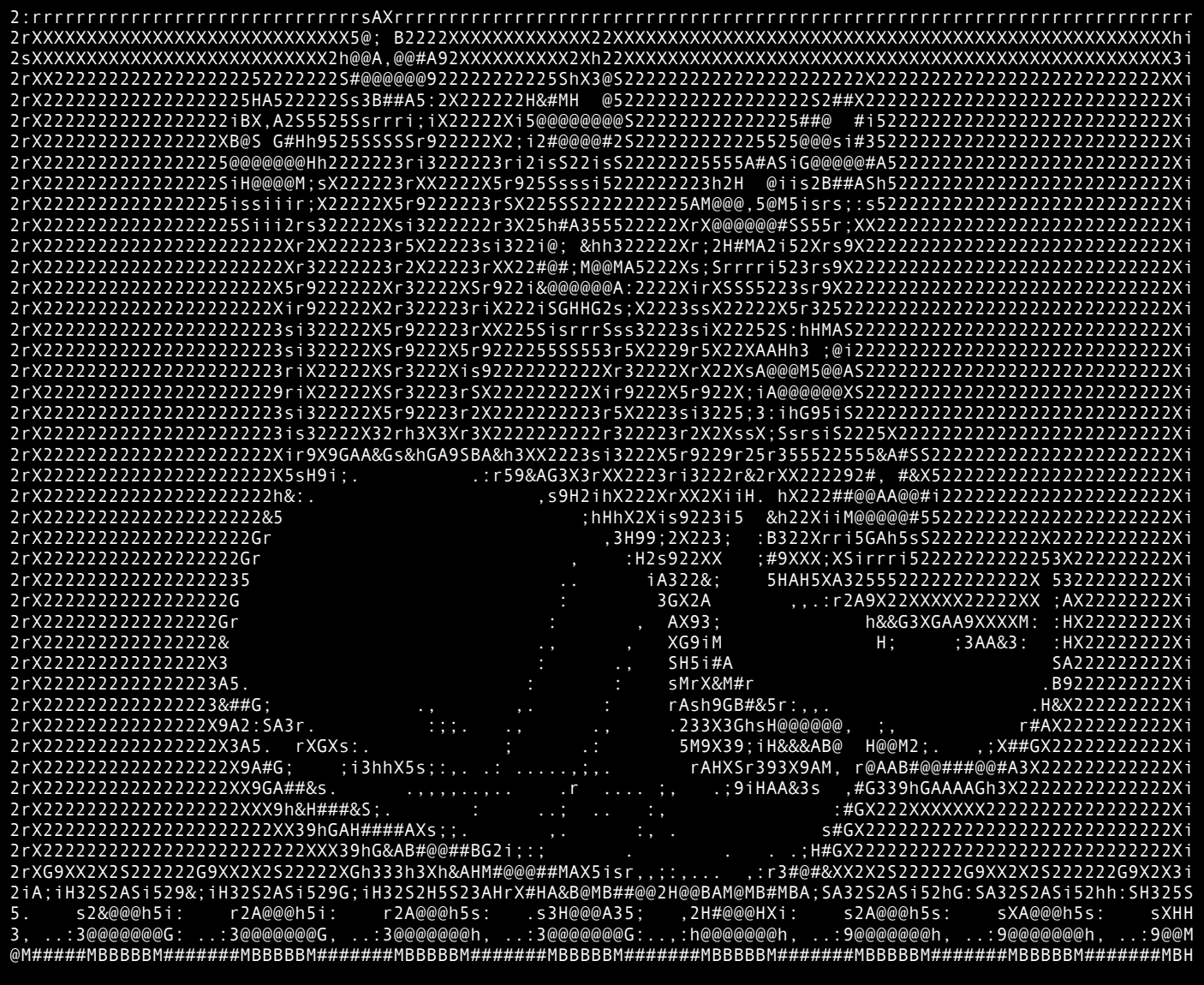
he’s a freak

$$2000 \text{ wpm} = 1.3 \text{ kbits/s}$$


2400 bps Modem

V22.bis = Blazing Fast

For 1988



Other Criticisms?

DONALD A. NORMAN

THE PSYCHOLOGY OF EVERYDAY THINGS

The Truth about UNIX:

the user interface is horrid (1981)

Names do not map to Functionality

High Cognitive Load

Feedback only occurs on Error

Inconsistent Syntax & Usage

He was absolutely correct

For 1981.

Bash 4.0

```
file.new
file.delete
file.properties

directory.new
directory.delete
directory.properties

...
```

```
alias file.new=”touch”
alias file.delete=”rm”
alias file.properties=”stat”

alias directory.new=”mkdir”
alias directory.delete=”rmdir”
alias directory.properties=”stat”

...
```

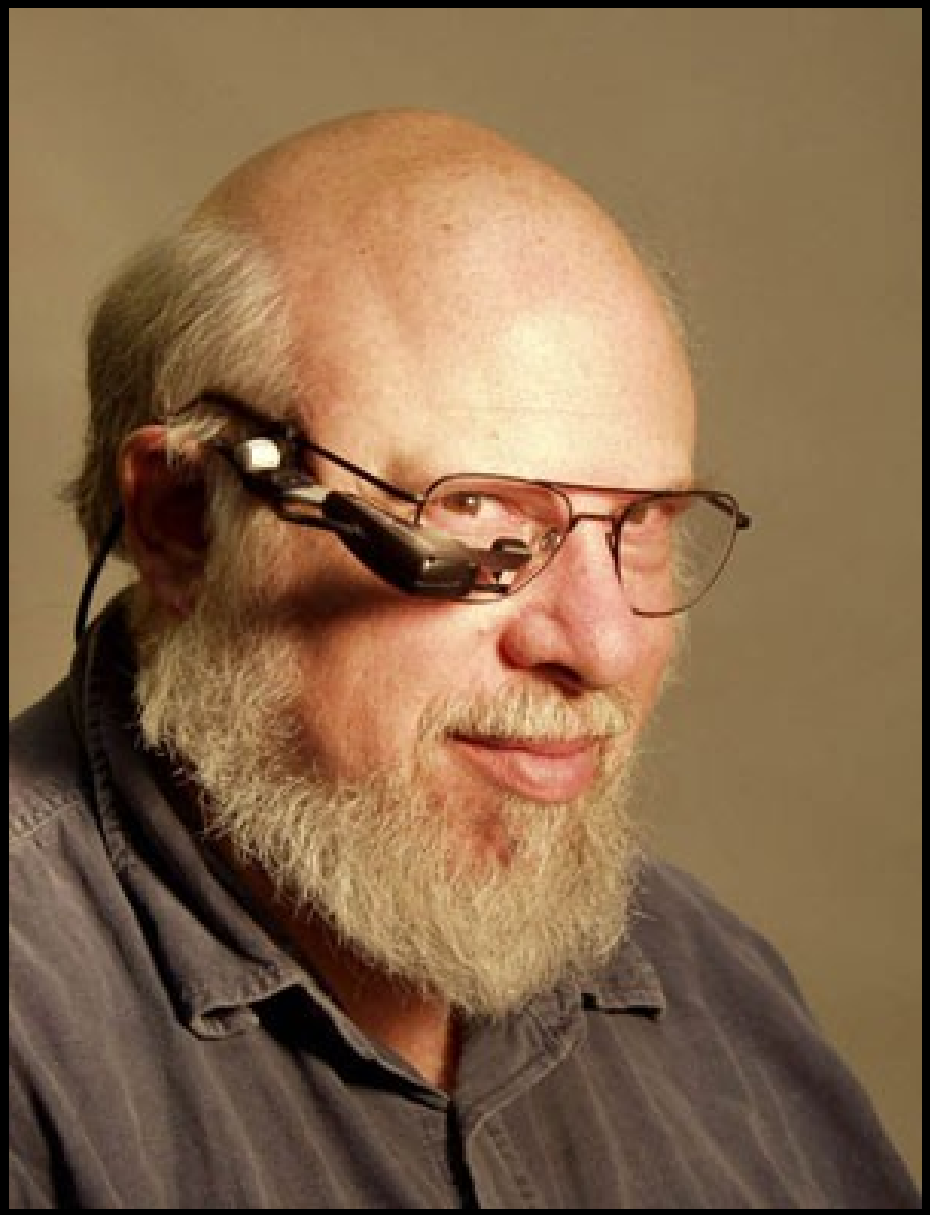
Mnemonics = Bash Completion

Alert Only On Error

He was wrong ;)

Written by Monkeys on Crack?

Snopes says True.



A computer shall not harm your work or, through inactivity, allow your work to come to harm.

```
function rm() {
  mv $* ${TMP_DIR}/.
}

set -C
```

A computer shall not waste your time or require you to do more work than is strictly necessary

Shannon's Law

$$\text{bits/s} = \text{frequency} * \log(1+\text{S/N})$$

N = our typing error rate

```
shopt -s dirspell
shopt -s cdspell
```

set completion-ignore-case on

To get any lazier,
we need to do some real work

Huffman Coding

Vary storage size inversely with frequency

```
sort .bash_history |
  awk '{ print $1 }' |
  uniq -c | sort -rn | less
```

Create tiny aliases
for the most common

```
me$ cd /var/tmp
me$ ls
```

```
cd() {
  builtin cd "$@"
  # list first bunch of entries
  directory.contents.wide | head
}
```

```
me$ chmod 0 lamers_file
me$ ls -l lamers_file
```

```
[[ ${PS1} = “” ]] && return

file_change() {
  IFS=
  $@
  shift
  for f in "$*" ; do
    if file.is_file "${f}"; then
      file.properties "${f}"
    fi
  done
}
```

```
mv()    { file_change
    command mv    "$@"; }

chmod() { file_change
    command chmod "$@"; }

chown() { file_change
    command chown "$@"; }

touch() { file_change
    command touch "$@"; }
```

Smart Prompts

know that attention is precious

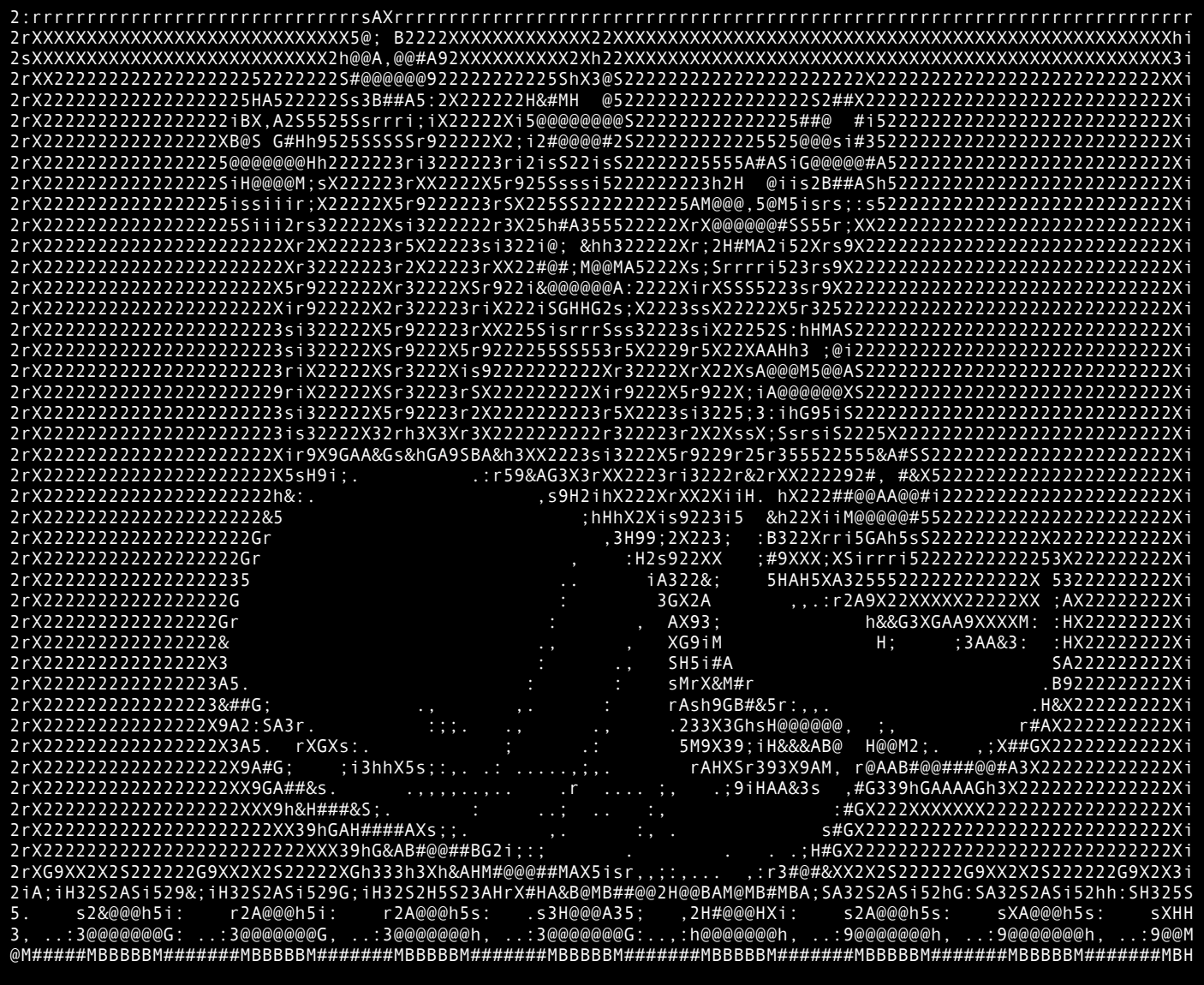
Seeing is believing?

```
sane_prompt() {
  export PROMPT_LAST_CMD_EXIT_CODE=$?
  history -a
  prompt_screen_shelltitle
  local pkts=$( prompt_pad
     ${TEXT_RED} prompt_network_loss )
  # wifi, directory size
  # SVN/CVS/GIT repo status
  # disk usage, cpu usage, …
  PS1="${hms}${box}${disk}${cpu}${wifi}${pkts}$
{dirsize}${repo}${err}${wd}"
  [[ ${#PS1} -gt ${PROMPT_MAX_LENGTH} ]] &&
     PS1="${PS1}\n"
  export PS1="${PS1}${sigil}${TEXT_BLACK} "
  export PROMPT_PREVIOUS_TIME=${SECONDS}
}
export PROMPT_COMMAND=sane_prompt
```

What about the Third Law?

```
sane_prompt() {
   ...
   # disabled by operator?
   [[ -e ${PROMPT_KILL_FILE} ]] && return

   # box too busy, be nice
   local load=$( cpu_usage_as_int )
   [[ ${load} -gt ${PROMPT_LOAD_MAX} ]] && {
      echo CPU LOAD ABOVE ${PROMPT_LOAD_MAX}
      return
   }
   ...
}
```



Towards a Smarter Bash

A Pattern Language

Towns · Buildings · Construction

Christopher Alexander
Sara Ishikawa · Murray Silverstein
WITH
Max Jacobson · Ingrid Fiksdahl-King
Shlomo Angel

```
echo 0.102.83.12.74.96 |
sed 's/\./ /g' |
each word printf '”%02x:”' {}
```

Any Ruby Fans Here?

```
each() {
  ...
  if [[ "$1" = "word" ]]; then
    shift
    while read line; do
      for word in ${line}; do
        eval "${@/\{\}/\"${word}\"}"
      done
    done
  fi
}
```

Completion = Speed

```
export TWITS=”${HOME}/.twits”
twit() {
  url="https://twitter.com/statuses/update.xml"
   __add() {
     grep -q ^$1\$ ${TWITS} ||
       echo $1 >> ${TWITS}
  }
  for word in $*; do
    [[ '@' = "${word/[^@]*/}" ]] && __add ${word}
  done
  output=$(curl -v -n -d status="$*" ${url} 2>&1)
  [[ $? != 0 ]] && echo ${output}
}
complete_twit() {
  local cur=${COMP_WORDS[COMP_CWORD]}
  COMPREPLY=( $( grep -i ${cur} ${TWITS} ) )
}
complete -F complete_twit twit
```

Future Work

```
# bash 4.0 error trapping
command_not_found_handle() {
   local cmd="$*"
   read -p "errors> " \
        -e \
        -i "${cmd}" cmd
   ${cmd}
}
```

Towards a Smarter Bash

Towards a Smarter Human

Thanks!

http://haller.ws/projects/bash