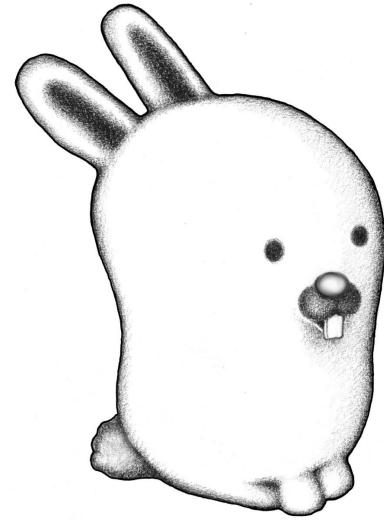


# Rc



**The Plan 9 OS Shell**

# Goals

- Improve on other UNIX shells
- Better parsing
- Be like Bourne/BASH but with "less idiosyncratic syntax"[\[1\]](#)
  - Syntax draws on C rather than Algol 68 (brackets rather than keywords)

```
if (test -f junk) rm junk
```

```
if test -f junk; then rm junk; fi
```

# Basics

```
date
```

```
cat /lib/news/build
```

```
who >user.names
```

```
rm -r junk || echo rm failed!
```

# Quoting

```
rm 'some file'
```

```
# with spaces
```

```
echo 'How' 's your father?'
```

```
# doubled apostrophes for lit.
```

# Patterns and Wildcards

```
ls foo* # foo_bar foo_baz foo
```

```
ls foo? # fool foot foo_
```

```
ls foo[a-z] # fool foot
```

```
ls foo[~a-z] # foo_
```

# Variables

```
# values are lists of strings
```

```
path=(. /bin)
```

```
# list of single string
```

```
user=td
```

```
font=/font/bit/pelm/ascii.9.font
```

# Variables continued

```
echo $path # => echo . /bin
```

```
echo $path(2) # => echo /bin
```

```
echo $path(2 1 2) # echo /bin .  
/bin
```

```
# note difference:
```

```
empty=()
```

```
null=''
```

# Arguments

`$*` # argument list

`$1` # same as `$* (1)`



# Concatenation

^ is the concat. operator

```
echo (a b c) ^ (1 2 3)
```

```
# => echo a1 b2 c3
```

```
src=(main subr io)
```

```
cc $src^.c
```

```
# => cc main.c subr.c io.c
```

# Use command output

```
cat `{ls -tr | sed 10q}
```

...output contents of ten most recent files  
(ascending)

(btw: that's a backtick, or leftquote, we lispers  
would say *quasiquote*)

# The gloves come off...

"Rc has syntax for some kinds of non-linear but treelike pipelines."[\[1\]](#)

```
cmp <{old} <{new}
```

...`cmp`'s input is `old`'s and `new`'s output in parallel

# Examples (1)

```
for(i) {  
    if(test -f /tmp/$i)  
        echo $i already in /tmp  
    if not  
        cp $i /tmp  
}
```

# Examples (2)

```
ps1='% '
tab='  '
fn cd{
    builtin cd $1 &&
    switch($#*) {
    case 0
        dir=$home
        prompt=($ps1 $tab)
```

```
        case *
            switch($1)
            case /*
                dir=$1
                prompt=(`{basename `{pwd}}
^$ps1 $tab)
            case */* *.*
                dir=()
                prompt=(`{basename `{pwd}}
^$ps1 $tab)
            case *
                dir=()
                prompt=($1^$ps1 $tab)
            }
        }
    }
```

# Examples (2)

```
fn pwd{  
    if (~ $#dir 0)  
        dir=`{/bin/pwd}`  
    echo $dir  
}
```

# There's more

- functions
- nice file descriptor redirection
- global and local variables

# References

1. <http://plan9.bell-labs.com/sys/doc/rc.html>
2. <https://en.wikipedia.org/wiki/Rc>
3. <http://plan9.bell-labs.com/magic/man2html/1/rc>
- 4.