The GNU Project and Me

27 Years with GNU Awk

Aharon (Arnold) Robbins
Author, Programmer
GNU Project Volunteer

arnold@skeeve.com

All Rights Reserved
The GNU Project and Me — 27 Years with GNU Awk

Agenda

• How I Got Involved
• Gawk Development Through The Years (seriously condensed)
• My Writing Career (also condensed)
• Lessons Learned Cloning A Unix Program
• The Current State of Awk
• Summary
How I Got Involved

• Involved in USENET since 1983; early and strong tradition of posting code and sharing
• 1987: Grey AWK book published
• Awk looked cool, but couldn’t get it
• GNU Project had a version — Great!
• At the time:
How I Got Involved

• Involved in USENET since 1983; early and strong tradition of posting code and sharing
• 1987: Grey AWK book published
• Awk looked cool, but couldn’t get it
• GNU Project had a version — Great!
• At the time:
  — I was single
How I Got Involved

• Involved in USENET since 1983; early and strong tradition of posting code and sharing
• 1987: Grey AWK book published
• Awk looked cool, but couldn’t get it
• GNU Project had a version — Great!
• At the time:
  — I was single
  — I had a home Unix System (AT&T 3B1)
How I Got Involved

- Involved in USENET since 1983; early and strong tradition of posting code and sharing
- 1987: Grey AWK book published
- Awk looked cool, but couldn’t get it
- GNU Project had a version — Great!
- At the time:
  - I was single
  - I had a home Unix System (AT&T 3B1)
  - I had a fair amount of time at work, too
1988: GNU Awk 1.x

- Written by Paul Rubin, Jay Fenlason, and Richard Stallman
- Clone of V7 awk, not the new language in the book
- Very buggy clone, at that:
  - Leaked memory copiously
  - Language bugs too. E.g., a missing action acted like an empty one!

```
/foo/       # No action, means { print }
/foo/ { }   # Empty action, do nothing
```

- No documentation
- David Trueman had already volunteered to bring gawk up to par
- I joined the fun. We coordinated via email (UUCP, no less!)
- We each had an RCS repository and we exchanged patches, only occasionally sending full tar files

- Added all `nawk` features
- Coordination with Brian Kernighan (then at Bell Labs) started. Still continues today
- Began involvement with POSIX standards
- Added a small number of enhancements, several of which went back into Brian Kernighan’s `nawk` and into POSIX
- Documentation included with the release (man page and Texinfo manual)
- Initial porting efforts: MS-DOS was the first port, VMS was the second
- Ad-hoc test suite started
- *Dark Corners*: With time, `gawk` continued to improve compatibility with Unix `nawk`. Performance also improved
- *No Arbitrary Limits* started to pay off: UUNET processes > 65,000 news articles for statistics gathering with a single program
- Mike Brennan’s `mawk` released
1993–1995: GNU Awk 2.15.x

- Focus on compatibility, portability, stability
- A modest number of new features
- More ports are done: Atari, Amiga, more Unix systems (including Cray!)
- `gawk` moves more into the mainstream as GNU/Linux starts to take off
- Late 1994 or 1995 I become the sole maintainer
- Shift to longer release cycles, with more patch releases
- Brian Kernighan’s `nawk` released through Netlib, later through his web page
1996–2000: GNU Awk 3.0.x

- Again, a modest number of new features
- Moved to GNU Autoconf. Lots of work, but big payoff
- More tests added to the test suite. Test suite organized somewhat
- Policy of (trying) not to add features in patches
- Stability continued to improve as usage went way up with GNU/Linux and BSD systems
- Development slowed more as my family grew and with our move to Israel at the end of 1997
2001–2010: GNU Awk 3.1.x

- Lots of new features (too many). Some highlights:
  - Coprocesses and TCP/IP networking
  - I18N support at the `awk` language level
  - `PROCINFO` array to replace `/dev/user` etc.
  - Bit manipulation and array sorting functions
- Statement level profiling with `pgawk`
- Test suite grows a lot, updating it simplified
- Many BSD systems move to `nawk` to avoid GPL software (Boo!)
- Made source code repository public, via CVS on `savannah.gnu.org`
  - 2006: At first via CVS, since I was comfortable with it
  - 2010: Moved to `git`, after urging from other GNU maintainers
- With move to `git`, put as many of the old releases as I could find into the repo, and also made them available from `ftp.gnu.org`
- General cadence of one release a year, minor features introduced with patch releases, no real progress on a major release for most of the time
2011: GNU Awk 4.0

- Released June, 2011
- Lots of new features (again):
  - Removed `/dev/pid` etc. special files (finally!)
  - POSIX behavior for `sub()` and `gsub()` (bad idea)
  - New `FPAT` variable and `patsplit()` function to describe field contents instead of separators
  - Enabled `switch` / `case` by default
  - New `BEGINFILE` and `ENDFILE` patterns
  - Short options to go with all long ones, for use with `#!`
  - Indirect function calls!
  - Support for IPv6!
  - Arrays of arrays!
  - Rational Range Interpretation (`[a-z]` means what G-d intended it should mean!)
2011: GNU Awk 4.0 (continued)

- Lots of new features (again) [continued]:
  - PROCINFO["sorted_in"] for sorted for loop traversal
  - Many code cleanups
  - Lots of other minor features and improvements
- New byte-code based internals give us a debugger!
2013: GNU Awk 4.1

- Released in May, 2013
- New features (but not quite as many):
  - Three executables (gawk, pgawk, and dgawk) merged into one.
  - Arbitrary precision arithmetic with MPFR.
  - New options −i and −l from XMLgawk project merged in.
  - Dynamic extension API totally redone and fully documented! Aims to provide source and binary compatibility going forward.
  - The and(), or(), and xor() functions take any number of arguments (minimumum 2)
  - New arrays PROCINFO["identifiers"], SYMTAB, and FUNCTAB.
My Writing Career

• In parallel with coding, I was trying to get paid for writing. Things progressed over time
My Writing Career

• In parallel with coding, I was trying to get paid for writing. Things progressed over time
• 1990 or 1991: Relationship with SSC started — KSH Reference Card
My Writing Career

• In parallel with coding, I was trying to get paid for writing. Things progressed over time
• 1990 or 1991: Relationship with SSC started — KSH Reference Card
• 1994-ish: Initial efforts at commercial publication of the Gawk Manual stymied by the FSF’s “anyone can copy” copying permissions (even O’Reilly!)
My Writing Career

- In parallel with coding, I was trying to get paid for writing. Things progressed over time
- 1990 or 1991: Relationship with SSC started — KSH Reference Card
- 1994-ish: Initial efforts at commercial publication of the Gawk Manual stymied by the FSF’s “anyone can copy” copying permissions (even O’Reilly!)
- 1994: SSC started Linux Journal. I wrote a few articles for them
My Writing Career

• In parallel with coding, I was trying to get paid for writing. Things progressed over time
• 1990 or 1991: Relationship with SSC started — KSH Reference Card
• 1994-ish: Initial efforts at commercial publication of the Gawk Manual stymied by the FSF’s “anyone can copy” copying permissions (even O'Reilly!)
• 1994: SSC started *Linux Journal*. I wrote a few articles for them
• 1996: SSC didn’t mind the FSF copying rules, so they published *Effective AWK Programming*, first and second editions, for versions 3.0.0 and 3.0.3 in 1996 and 1997
My Writing Career

- In parallel with coding, I was trying to get paid for writing. Things progressed over time.
- 1990 or 1991: Relationship with SSC started — KSH Reference Card.
- 1994-ish: Initial efforts at commercial publication of the Gawk Manual stymied by the FSF’s “anyone can copy” copying permissions (even O’Reilly!)
- 1994: SSC started *Linux Journal*. I wrote a few articles for them.
- 1996: SSC didn’t mind the FSF copying rules, so they published *Effective AWK Programming*, first and second editions, for versions 3.0.0 and 3.0.3 in 1996 and 1997.
- 1996: O’Reilly wasn’t interested in replacing the “awk” part of *sed & awk* with the *gawk* documentation or publishing it separately.
My Writing Career

- In parallel with coding, I was trying to get paid for writing. Things progressed over time
- 1990 or 1991: Relationship with SSC started — KSH Reference Card
- 1994-ish: Initial efforts at commercial publication of the Gawk Manual stymied by the FSF’s “anyone can copy” copying permissions (even O’Reilly!)
- 1994: SSC started Linux Journal. I wrote a few articles for them
- 1996: SSC didn’t mind the FSF copying rules, so they published Effective AWK Programming, first and second editions, for versions 3.0.0 and 3.0.3 in 1996 and 1997
- 1996: O’Reilly wasn’t interested in replacing the “awk” part of sed & awk with the gawk documentation or publishing it separately
- However, sed & awk needed revising. They asked me to do it and I did, since it really needed it. The second edition came out early in 1997
My Writing Career

• In parallel with coding, I was trying to get paid for writing. Things progressed over time
• 1990 or 1991: Relationship with SSC started — KSH Reference Card
• 1994-ish: Initial efforts at commercial publication of the Gawk Manual stymied by the FSF’s “anyone can copy” copying permissions (even O'Reilly!)
• 1994: SSC started Linux Journal. I wrote a few articles for them
• 1996: SSC didn’t mind the FSF copying rules, so they published Effective AWK Programming, first and second editions, for versions 3.0.0 and 3.0.3 in 1996 and 1997
• 1996: O’Reilly wasn’t interested in replacing the “awk” part of sed & awk with the gawk documentation or publishing it separately
• However, sed & awk needed revising. They asked me to do it and I did, since it really needed it. The second edition came out early in 1997
• Surprise! $$ $$
My Writing Career (continued)

- Late in 1997, knowing that I was moving and would not find a “real” job right away, I took on revisions of two more O’Reilly books
My Writing Career (continued)

• Late in 1997, knowing that I was moving and would not find a “real” job right away, I took on revisions of two more O’Reilly books.
• After finding that Israeli companies are like American ones, looking for 60-90 hour work weeks, I decided to try writing full time.
My Writing Career (continued)

• Late in 1997, knowing that I was moving and would not find a “real” job right away, I took on revisions of two more O’Reilly books
• After finding that Israeli companies are like American ones, looking for 60-90 hour work weeks, I decided to try writing full time
• Published a number of books with O’Reilly, including the gawk manual in 2001
My Writing Career (continued)

• Late in 1997, knowing that I was moving and would not find a “real” job right away, I took on revisions of two more O’Reilly books

• After finding that Israeli companies are like American ones, looking for 60-90 hour work weeks, I decided to try writing full time

• Published a number of books with O’Reilly, including the gawk manual in 2001

• Also published a book on Linux programming with Prentice Hall.
My Writing Career (continued)

• Late in 1997, knowing that I was moving and would not find a “real” job right away, I took on revisions of two more O’Reilly books.
• After finding that Israeli companies are like American ones, looking for 60-90 hour work weeks, I decided to try writing full time.
• Published a number of books with O’Reilly, including the *gawk* manual in 2001.
• Also published a book on Linux programming with Prentice Hall.
• Being independent lasted until 2005 when I had to get a “real” job.
My Writing Career (continued)

• Late in 1997, knowing that I was moving and would not find a “real” job right away, I took on revisions of two more O’Reilly books.
• After finding that Israeli companies are like American ones, looking for 60-90 hour work weeks, I decided to try writing full time.
• Published a number of books with O’Reilly, including the `gawk` manual in 2001.
• Also published a book on Linux programming with Prentice Hall.
• Being independent lasted until 2005 when I had to get a “real” job.
• I continue to write, just at a slower rate.
My Writing Career (continued)

• Late in 1997, knowing that I was moving and would not find a “real” job right away, I took on revisions of two more O'Reilly books
• After finding that Israeli companies are like American ones, looking for 60-90 hour work weeks, I decided to try writing full time
• Published a number of books with O'Reilly, including the *gawk* manual in 2001
• Also published a book on Linux programming with Prentice Hall.
• Being independent lasted until 2005 when I had to get a “real” job
• I continue to write, just at a slower rate
• With any luck, O'Reilly to publish an updated *gawk* manual – WIP
My Writing Career (continued)

• Late in 1997, knowing that I was moving and would not find a “real” job right away, I took on revisions of two more O’Reilly books
• After finding that Israeli companies are like American ones, looking for 60-90 hour work weeks, I decided to try writing full time
• Published a number of books with O’Reilly, including the **gawk** manual in 2001
• Also published a book on Linux programming with Prentice Hall.
• Being independent lasted until 2005 when I had to get a “real” job
• I continue to write, just at a slower rate
• With any luck, O’Reilly to publish an updated **gawk** manual – WIP
• With more luck, Prentice Hall to publish an updated edition of my Linux programming book – WIP
My Writing Career (continued)

- Late in 1997, knowing that I was moving and would not find a “real” job right away, I took on revisions of two more O’Reilly books
- After finding that Israeli companies are like American ones, looking for 60-90 hour work weeks, I decided to try writing full time
- Published a number of books with O’Reilly, including the `gawk` manual in 2001
- Also published a book on Linux programming with Prentice Hall.
- Being independent lasted until 2005 when I had to get a “real” job
- I continue to write, just at a slower rate
- With any luck, O’Reilly to publish an updated `gawk` manual – WIP
- With more luck, Prentice Hall to publish an updated edition of my Linux programming book – WIP
- Search for “Arnold Robbins” on Amazon.com to get a full list
Lessons Learned Cloning A Unix Program

_Dark corners are basically fractal — no matter how much you illuminate, there’s always a smaller but darker one._

— Brian Kernighan

•
Lessons Learned Cloning A Unix Program

Dark corners are basically fractal — no matter how much you illuminate, there’s always a smaller but darker one.

— Brian Kernighan

- Good documentation isn’t always enough; the grey AWK book doesn’t specify every last picayune detail
Lessons Learned Cloning A Unix Program

Dark corners are basically fractal — no matter how much you illuminate, there’s always a smaller but darker one.

— Brian Kernighan

• Good documentation isn’t always enough; the grey AWK book doesn’t specify every last picayune detail
• Checking program behavior doesn’t always help either, sometimes the program is wrong!
Lessons Learned Cloning A Unix Program

*Dark corners are basically fractal — no matter how much you illuminate, there’s always a smaller but darker one.*

— Brian Kernighan

- Good documentation isn’t always enough; the grey AWK book doesn’t specify every last picayune detail
- Checking program behavior doesn’t always help either, sometimes the program is wrong!
- Lots of dark corners to be explored. > 30 listed in EAP4 index
Lessons Learned Cloning A Unix Program

Dark corners are basically fractal — no matter how much you illuminate, there’s always a smaller but darker one.

— Brian Kernighan

• Good documentation isn’t always enough; the grey AWK book doesn’t specify every last picayune detail
• Checking program behavior doesn’t always help either, sometimes the program is wrong!
• Lots of dark corners to be explored. > 30 listed in EAP4 index
• Lots of brainos. For example:

  ```
  function foo(foo, bar) { c = foo }
  ```

  This gave a core dump in `gawk`, infinite recursion and core dump in `nawk` (both since fixed)
Lessons Learned Cloning A Unix Program

Dark corners are basically fractal — no matter how much you illuminate, there’s always a smaller but darker one.

— Brian Kernighan

• Good documentation isn’t always enough; the grey AWK book doesn’t specify every last picayune detail
• Checking program behavior doesn’t always help either, sometimes the program is wrong!
• Lots of dark corners to be explored. > 30 listed in EAP4 index
• Lots of brainos. For example:

  function foo(foo, bar) { c = foo }

  This gave a core dump in gawk, infinite recursion and core dump in nawk (both since fixed)
• Compatibility requires supporting silly things. E.g., in earlier versions of nawk outside a loop, the break and continue keywords act like next
Lessons Learned Cloning A Unix Program (cont.)

- No-one reads the documentation (sigh)
Lessons Learned Cloning A Unix Program (cont.)

• No-one reads the documentation (sigh)
• Be careful of creeping featurism:
  — Easy to add features, but hard to do it well (*IGNORECASE* took years to get right!)
  — Hard to remove bad ideas
  — Leads to portability and compatibility issues
  — Code base grows incrementally, such that you don’t notice that the program is too big to maintain well
  — My feeling is that many neat features just aren’t used
Lessons Learned Cloning A Unix Program (cont.)

• No-one reads the documentation (sigh)
• Be careful of creeping featurism:
  — Easy to add features, but hard to do it well (\texttt{IGNORECASE} took \textit{years} to get right!)
  — Hard to remove bad ideas
  — Leads to portability and compatibility issues
  — Code base grows incrementally, such that you don’t notice that the program is too big to maintain well
  — My feeling is that many neat features just aren’t used

• Awk wasn’t really designed for extensibility
Lessons Learned Cloning A Unix Program (cont.)

• No-one reads the documentation (sigh)
• Be careful of creeping featurism:
  — Easy to add features, but hard to do it well (IGNORECASE took years to get right!)
  — Hard to remove bad ideas
  — Leads to portability and compatibility issues
  — Code base grows incrementally, such that you don’t notice that the program is too big to maintain well
  — My feeling is that many neat features just aren’t used

• Awk wasn’t really designed for extensibility
• Test, test, and retest. Redesign code carefully! E.g., a nasty per-record memory leak in versions 3.0.1 and 3.0.2
Personal Benefits

• Fan mail: “Gawk is great, it just works when Unix awk falls over.” (Less of this today, lots early on)
Personal Benefits

- Fan mail: “Gawk is great, it just works when Unix awk falls over.” (Less of this today, *lots* early on)
- Interaction and/or friendships with famous people:
  - Brian Kernighan (the “K” in “K & R”)
  - Mike Brennan, author of *mawk*
  - Richard Stallman (less so)
Personal Benefits

- Fan mail: “Gawk is great, it just works when Unix awk falls over.” (Less of this today, lots early on)
- Interaction and/or friendships with famous people:
  - Brian Kernighan (the “K” in “K & R”)
  - Mike Brennan, author of mawk
  - Richard Stallman (less so)
- Friendships with other GNU maintainers, both via email and/or in person (e.g., at USENIX conferences)
Personal Benefits

• Fan mail: “Gawk is great, it just works when Unix awk falls over.” (Less of this today, *lots* early on)
• Interaction and/or friendships with famous people:
  — Brian Kernighan (the “K” in “K & R”)
  — Mike Brennan, author of mawk
  — Richard Stallman (less so)

• Friendships with other GNU maintainers, both via email and/or in person (e.g., at USENIX conferences)
• Occasionally get to give talks at local conferences :-)
Professional Benefits

- Spent many years as an author and contract programmer. This sure beat 80-hour work weeks and a two-hour daily commute.
- O’Reilly work started with revision of *sed & awk*, as direct result of being *gawk* maintainer.
- Now have multiple O’Reilly books to my name. In 1995, I would have never expected to see my name on *Unix in a Nutshell*.
- Friendship with Mike Brennan led to some contract programming where I learned quite a lot.
- As a result of being an O’Reilly author, wrote a book for Prentice Hall and am also a series editor.
- A long-time Israeli friend through *gawk* work helped me get a position in Industry (in Tel Aviv) in 2005.
- (Since 2008, working in Jerusalem)
Current State of the World — Awk

- **nawk** source released in 1993 via netlib, not publicized. Available from Brian Kernighan’s home page, since late 1990s. **nawk** has improved a lot because of this. Most BSDs use it

- **mawk** is fast, stable, and well written. Mike Brennan stopped working on it in 1996, but in ~2009 Thomas Dickey took over maintenance

- **gawk** is de-facto standard on GNU/Linux
  - Today, pretty stable. Dark corners are fewer and smaller
  - Often, **gawk** is correct and others aren’t. E.g.:
    ```c
    printf("%d\n", 2 ^ 40)
    ```
  - **gawk --posix** is the most POSIX compliant version that I know of, especially among free Awks
  - Only one with networking, loadable functions in C
  - Has the most built-in variables and functions. This is either a bug or a feature
  - The byte-code internals offer potential speedup, may one day match **mawk** in speed
Summary

• Initial **gawk** work focused on compatibility with Unix **nawk** and reasonable performance

• Later, POSIX compliance became important:
  — Getting the standard itself correct
  — Making **gawk** conform to it
  — Helping the other Awks conform too

• Over time, *feeping creaturism* has set in. I don’t want the **gawk** language to grow any more (not succeeding at this)

• Unfortunately, adding “just one more feature” is addictive

• Having new blood in the project has been good!

• Development is slower than I’d like, but that’s life

• Making the source repository public was a good idea!

• **Doing Free Software is both fun and important**
Questions and Answers
# Contents

- The GNU Project and Me ..................................................... 1
- 27 Years with GNU Awk ....................................................... 1
- Agenda ................................................................. 2
- How I Got Involved ............................................................. 3
- 1988: GNU Awk 1.x .............................................................. 4
- 1993–1995: GNU Awk 2.15.x ............................................... 6
- 1996–2000: GNU Awk 3.0.x ................................................. 7
- 2001–2010: GNU Awk 3.1.x ............................................... 8
- 2011: GNU Awk 4.0 ............................................................. 9
- 2011: GNU Awk 4.0 (continued) ........................................ 10
- 2013: GNU Awk 4.1 ........................................................... 11
- My Writing Career .............................................................. 12
- My Writing Career (continued) ........................................... 13
- Lessons Learned Cloning A Unix Program ......................... 14
- Lessons Learned Cloning A Unix Program (cont.) .......... 15
- Personal Benefits ............................................................. 16
- Professional Benefits ......................................................... 17
- Current State of the World — Awk .................................. 18
- Summary ................................................................. 19
- Questions and Answers ..................................................... 20