

### Seminar Origins of Operating Systems





# Mario Oschwald - mario.oschwald@hpi.uni-potsdam.de

Operating Systems and Middleware – Prof. Polze Hasso-Plattner-Institute





- □ History and Background
- □ Plan 9's Characteristics

🗆 Demo







- □ UNIX was developed in the early 70's
- □ Some of its characteristics:
  - Everything is a file even system resources can be accessed via basic file system calls (open, close, read, write)
  - Multiuser different people can user a computer at the same time
  - Multitasking the computer can do several things at the same time
- Originally UNIX ran on centralized systems to which users where connected directly using terminals
- In the middle 80's the trend towards decentralized systems was very strong
- People were now using workstations just for themselves that were connected through a network
- → Not everybody liked this development...





- $\Box$  UNIX slowly moved away from the "everything is a file" mantra
- Objections to decentralized workstations:
  - High administrative costs administering many workstations is expensive
  - Waste of resources memory, cpu and disk space are not used efficiently
  - Location dependence the same environment is not available at every workstation
  - Alienation losing the feeling of working together on a system, because everyone has their own system
- Some of the original UNIX developers wanted to rectify these issues while keeping the advantages of UNIX:
  - "In building Plan 9, we generalized proven ideas from the Unix operating system rather than add new untried concepts"[1]
- $\Box$  And so Plan 9 was conceived



## **People behind Plan 9**



# □ Ken Thompson

- one of the original Unix developers
- developer of C
- Rob Pike
- $\Box$  Russ Cox
  - current maintainer of Plan 9
- $\Box$  Many others:
  - Dennis Ritchie
  - Dave Presotto
  - Tom Duff





### **Release History**



Developed at the Bell Labs

Given Four releases:

- 1<sup>st</sup> edition (1992) only available to universities
- 2<sup>nd</sup> edition (1995) available in book and CD form for \$350
- 3<sup>rd</sup> edition (2000) freely available over the Internet
- 4<sup>th</sup> edition (2002) free software
- Still a lightweight system with moderate resource requirements:
  - 18000 LOC overall
  - 5000 LOC for basic kernel
  - Rest for file servers
- Actively maintained (~10-15 mails/day in mailing lists)
- Commercial offspring Inferno (sold) by Vita Nuova
- □ Some influences on other operating systems (BSD, Linux)



#### The Three Plan 9 Mantras



- Everything is a file
  - "resources are named and accessed like files in a hierarchical file system"[2]
- $\Box$  Every file can be accessed independently of location
  - "there is a standard protocol, called 9P, for accessing these resources"[2]
  - Every process has a unique name space
    - "the disjoint hierarchies provided by different services are joined together into a single private hierarchical file name space"[2]
    - name space in this context means a view of a file system hierarchy



# Everything is a file



Device and network access purely file system based:

- /dev/mouse, /dev/cons, /dev/screen mouse, console, monitor
- /net/etherN, /net/tcp, /net/upd, /net/icmp network access
- /dev/sd\* disks and cdroms
- /dev/ei\* serial lines
- $\Box\,$  /proc file system later copied by Linux and BSD
  - Enivronment variables available through /env
  - /dev/snarf clipboard (only text sorry!)
  - User space file servers:
    - ftp
    - mail
    - window system
    - name services





9

- $\Box$  All file servers (remote or local) are accessed using 9P
- □ Stateful
- Local access through function calls, remote access originally via an own transport protocol, IL, today TCP
- $\Box$  Kernel multiplexes between local, remote and in kernel file servers
- $\Box$  9P messages (and functions)[3]:
- version start a new session auth – optionally authenticate subsequent attaches attach – attach to the root of a file tree walk – walk up or down in Operating Systems and Modeleware = Prof. Polze Hasso-Platmer-Institute Open – open a file
- create create a new file read – read data from an open file write – write data to an open file clunk – discard a file tree reference (close) remove – remove a file stat – retrieve file's attributes wstat – set a file's attribute



#### **Unique Namespaces**



 $\Box$  Built using two basic system calls:

- mount make a resource from outside the current name space available
- bind make a file from the current name space available in a different location
- Bootstrapped via special kernel file servers
- □ Processes can share, inherit or start with an empty name space
- Union directories possible through mounting directories at the same location
- Current name space available through /proc/\$pid/ns





□ Plan 9 supports several architectures (MIPS,SPARC, Alpha, Intel x86...)

- □ mk used as a make replacement
- □ No #if support in preprocessor
- $\Box$  Process creation using fork() or the new rfork() system call:
  - RFNOWAIT child process does not send exit message
  - RFNAMEG/RFCNAMEG [copy the parent|start with an empty] name space
  - RFENVG/RFCENVG [copy parent|start with an empty] environment
  - RFMEM share data segments with parent
- Separate thread library available (besides RFMEM) but scheduled cooperatively (no non-blocking system calls)
- □ Signal delivery text based (notes: /proc/\$pid/note)
- Unique synchronization mechanism: rendezvous()
- Posix compatible environment (APE) available





- □ No superuser
- Uids not numeric
- $\hfill\square$  UTF-8 was developed for Plan 9
- $\Box$  Signals are text based
- □ No soft/hard links
- $\Box$  No ioctls
- $\Box$  40 system calls vs. 300 in Linux



## Plan 9 Demo



Thu <sup>N</sup> Jul 6 06:06		
New Cut Paste Snarf Sort Z		New Cut Paste Snarf Sort Zerox De
Vusr/glenda/ Del Snarf Get readme.acme Del Snarf   Look Rio is the Plan 9 window system.	id. Acme is a s, browse the	■ [/usr/glenda/- Del Snarf   Look Sen term% [
To read more of this window, the up and down arrows scroll the text up and down half screens.	e into id down	
To effectively use rio, you need at least a three button mouse. If you only have a two button mouse you can emulate the middle button by holding down shift key whilst pressing the right button.	into two in blue). The xt:	
Button 1, 2, and 3 are used to refer to the left, middle, and right buttons respectively.	contains	
THE POP-UP MENU		
Pressing and holding down button 3 on the desktop or	fferent	/mail/fs/mbox/ Del Snarf   Look P
<ul> <li>Button 1 can be used to select text (press it, sweep, release it), and also to select the point where text would be inserted in the window. Use to the point where text would be inserted in the window. We Button 2 can be used to execute things. For example, use button 1 to type "Is -I" before "lib/" in the window showing /usr/glenda. Now use button 2 to select "Is -I lib/" (press it, select, release it). As you can see, button 2 means "execute this".</li> <li>Button 3 can be used to get things. For example, click button 3 on "lib/" within the "lusr/glenda" window. Can you see how a new window shows the contents of "Jusr/glenda" window. Just click button 3 on the thing you want to search within the body of a window. Just click button 3 on the thing you want to search. Again, you can selection.</li> <li>You can double-click with button 1 to select words; a double click at the end or beginning of a line selects the whole line. Once you have text selected text. A single click of button 2 would execute the word clicked as a command.</li> <li>Now let's pay attention to the tag line once more. As you can see, the left part has a path. That is the name for the window and shows</li> </ul>		13 Apr 2002 welcome!





- Innovative concepts, implemented technically excellent, some found their way into other operating systems
- Very lightweight, well structured kernel and free software, so possible use in teaching
- Development more ideologically driven than through pragmatics (How to do it more important than what to do with it)
- □ Not much end user uptake, users mainly researchers
- Limited set of programming languages and software libraries (No java, python, perl, ruby, GUI toolkits)
- Possible use in grid scenarios because of well integrated network transparency, authentication domains, service discovery[4]

No web browser

□ Not enough games





- 1) D. Presotto et al. "Plan 9, A Distributed System" http://citeseer.ist.psu.edu/presotto91plan.html
- 2) R. Pike et al., "Plan 9 from Bell Labs", http://plan9.bell-labs.com/sys/doc/9.pdf
- 3) C.H. Forsyth, "The Ubiquitous File Server in Plan 9", http://medias.2005.rencontresmondiales.org/topics/os/papers/forsyth-plan-9.pdf
- 4) A. Mirtchovsky, R. Simmonds, R. Minnic, "Plan 9 an Integrated Approach to Grid Computing", 2004, http://plan9.aichi-u.ac.jp/9grid/plan9-grid.pdf
- 5) R. Pike, K. Thompson "Hello World ...", http://plan9.bell-labs.com/sys/doc/utf.pdf
- 6) S. Mullender, D. Presotto, "Programming Distributed Applications usign Plan 9 from Bell Labs", http://www.cs.unibo.it/ersads/tutorials/mullender.ps
- 7) R. Pike, "Lexical File Names in Plan 9 or Getting Dot-Dot Right", http://plan9.bell-labs.com/sys/doc/lexnames.pdf
- 8) R. Cox et. al, "Security in Plan 9", http://plan9.bell-labs.com/sys/doc/auth.pdf





- 1) D. Presotto, P. Winterbottom, "The Organization of Networks in Plan 9", http://cm.bell-labs.com/sys/doc/net.pdf
- 2) R. Pike et al. "The Use of Name Spaces in Plan 9", http://cm.bell-labs.com/sys/doc/names.pdf
- "Andy Tanenbaum hasn't learned anything" a very interesting email from the Plan 9 developers dated '92, http://fred.cambridge.ma.us/c.o.r.flame/msg00025.html
- 4) "UTF-8 history" the real story how UTF-8 was developed http://www.cl.cam.ac.uk/~mgk25/ucs/utf-8-history.txt
- 5) No browser: http://cm.bell-labs.com/plan9/about.html "Because Plan 9 has a very different system model from other modern operating systems, it is sometimes difficult to port external software to Plan 9. In particular, Plan 9 has no full-featured web browser; webfs(4) and html(2) are intended as steps toward a solution."