

Recording all Output from a Student Radio Station

LAC2005

John ffitch and Tom Natt



UNIVERSITY OF
BATH



Introduction

Legal requirements for small radio stations in the UK mean, *inter alia*, that the student station at Bath (University Radio Bath or URB) must retain about 50 days of the station's output.

In addition, as it has recently become easier to transfer data using disposable media, and general technical savvy amongst presenters has improved, there is now some interest in producing personal archives of radio shows.

This presentation describes our solution.

The Regulation

- **You are required to make a recording of all broadcast output, including advertisements and sustaining services. You must retain these recordings ('logging tapes') for a period of 42 days after broadcast, and make them readily available to us or to any other body authorised to deal with complaints about broadcast programmes. Failure to provide logging tapes on request will be treated seriously, and may result in a sanction being imposed.**

The Old System

The Old System

- Logging was undertaken using a video player and a pile of video tapes.

The Old System

- Logging was undertaken using a video player and a pile of video tapes.
- These tapes were cycled manually so there was a single continuous recording of all output.

The Old System

- Logging was undertaken using a video player and a pile of video tapes.
- These tapes were cycled manually so there was a single continuous recording of all output.
- The quality was largely unknown — not even sure they work!

The Old System

- Logging was undertaken using a video player and a pile of video tapes.
- These tapes were cycled manually so there was a single continuous recording of all output.
- The quality was largely unknown — not even sure they work!
- The system required someone to physically change the tape, and this was often forgotten.

The Old System

- Logging was undertaken using a video player and a pile of video tapes.
- These tapes were cycled manually so there was a single continuous recording of all output.
- The quality was largely unknown — not even sure they work!
- The system required someone to physically change the tape, and this was often forgotten.
- Even if the tapes worked there was no mechanism for copying the tapes, certainly without removing logging.

end

A Shinning New Solution



has been moving to a computer-based station, so naturally the question was how easy would it be to incorporate logging.
But....

A Shinning New Solution



has been moving to a computer-based station, so naturally the question was how easy would it be to incorporate logging.
But....

- A financial crisis meant that there was no budget.

A Shinning New Solution



has been moving to a computer-based station, so naturally the question was how easy would it be to incorporate logging.
But....

- A financial crisis meant that there was no budget.
- Chance questions led to the design below.

end

Requirements

Requirements

- Should run reliably for many days, or even years, with little or no manual intervention.

Requirements

- Should run reliably for many days, or even years, with little or no manual intervention.
- Should log all audio output from the radio station.

Requirements

- Should run reliably for many days, or even years, with little or no manual intervention.
- Should log all audio output from the radio station.
- Maintain at least 50 days of material.

Requirements

- Should run reliably for many days, or even years, with little or no manual intervention.
- Should log all audio output from the radio station.
- Maintain at least 50 days of material.
- Able to recover any particular section of audio by time (and by a non-technical user).

Requirements

- Should run reliably for many days, or even years, with little or no manual intervention.
- Should log all audio output from the radio station.
- Maintain at least 50 days of material.
- Able to recover any particular section of audio by time (and by a non-technical user).
- Cost as close to zero as possible!

end

Calculations

Calculations

- Assuming input is at CD quality we need $44100 \times 2 \times 2 \times 60 \times 60$ bytes a day.

Calculations

- Assuming input is at CD quality we need $44100 \times 2 \times 2 \times 60 \times 60$ bytes a day.
- That is approximately 14Gb/day or 700Gb in 50 days.

Calculations

- Assuming input is at CD quality we need $44100 \times 2 \times 2 \times 60 \times 60$ bytes a day.
- That is approximately 14Gb/day or 700Gb in 50 days.
- That is beyond our budget so compression is necessary.

Calculations

- Assuming input is at CD quality we need $44100 \times 2 \times 2 \times 60 \times 60$ bytes a day.
- That is approximately 14Gb/day or 700Gb in 50 days.
- That is beyond our budget so compression is necessary.
- Assuming 1:8 compression we need at least an 80Gb disk, and experiments suggested a 400MHz Intel processor; I assumed Linux without thought.

Calculations

- Assuming input is at CD quality we need $44100 \times 2 \times 2 \times 60 \times 60$ bytes a day.
- That is approximately 14Gb/day or 700Gb in 50 days.
- That is beyond our budget so compression is necessary.
- Assuming 1:8 compression we need at least an 80Gb disk, and experiments suggested a 400MHz Intel processor; I assumed Linux without thought.
- Software base – C, shell scripts, cronjobs and PERL

end

Hardware

Hardware

- We scavenged a 550MHz Celeron with 128Mb, also ethernet and two old Soundblasters.

Hardware

- We scavenged a 550MHz Celeron with 128Mb, also ethernet and two old Soundblasters.
- Installed SuSE9.1 with borrowed keyboard, screen and mouse.

Hardware

- We scavenged a 550MHz Celeron with 128Mb, also ethernet and two old Soundblasters.
- Installed SuSE9.1 with borrowed keyboard, screen and mouse.
- Spent money on a new 120Gb disk.

end

Problems

Problems

- The disk delivered was faulty and had to be returned.

Problems

- The disk delivered was faulty and had to be returned.
- A small fire in the computer destroyed the processor (but luckily not the disk).

Problems

- The disk delivered was faulty and had to be returned.
- A small fire in the computer destroyed the processor (but luckily not the disk).
- Second scavenged machine was a 433MHz Celeron, but also more memory.

Problems

- The disk delivered was faulty and had to be returned.
- A small fire in the computer destroyed the processor (but luckily not the disk).
- Second scavenged machine was a 433MHz Celeron, but also more memory.
- The dead-line for the 2004 academic year was getting close.....

end

The Software Design

The Software Design

- Need a continuous reading of the audio, and periodic compression.

The Software Design

- Need a continuous reading of the audio, and periodic compression.
- After experimentation we created a threaded program, with threads to read audio, file audio, compress data, and other tasks.

The Software Design

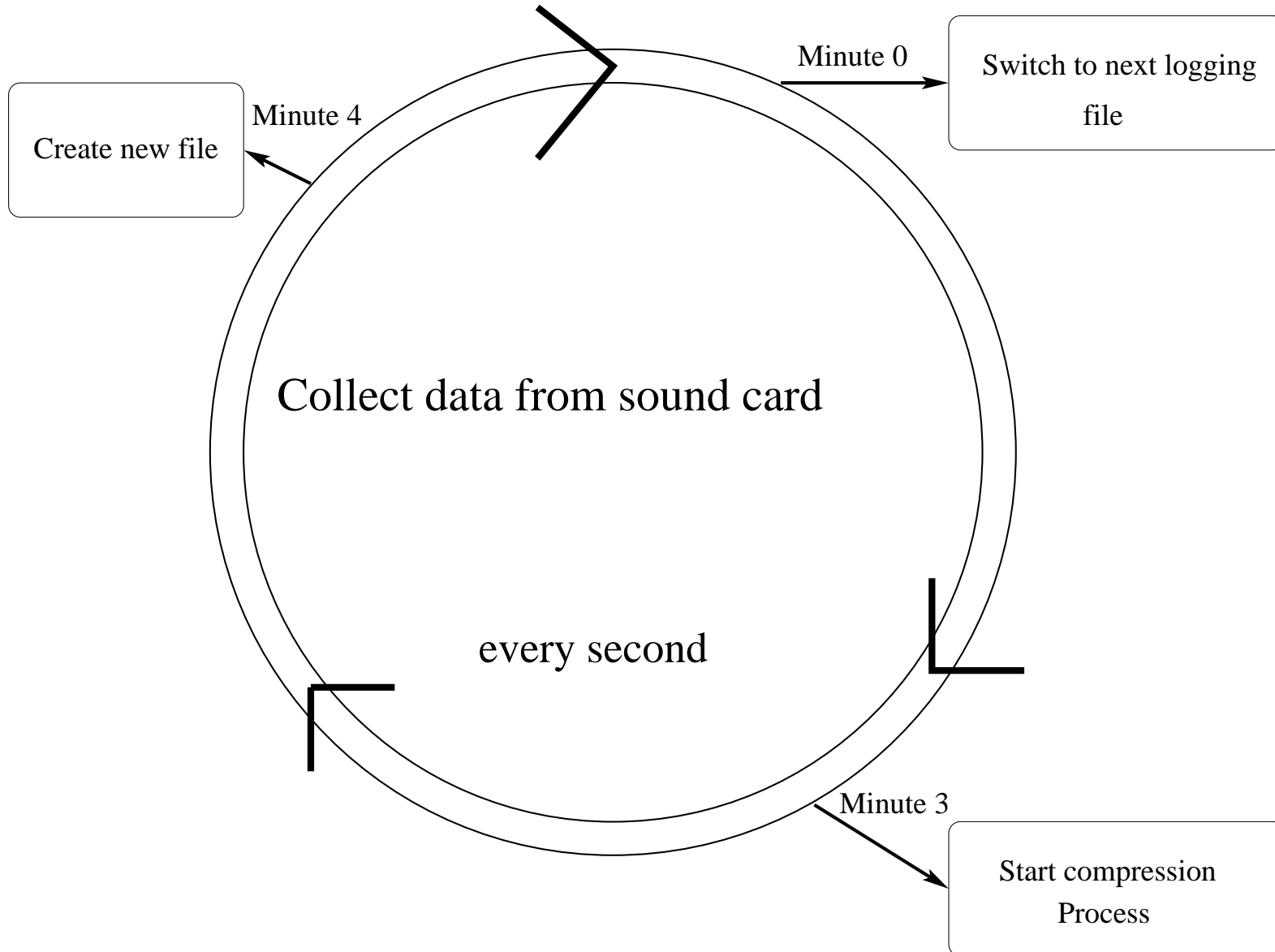
- Need a continuous reading of the audio, and periodic compression.
- After experimentation we created a threaded program, with threads to read audio, file audio, compress data, and other tasks.
- Largely based on the PortAudio test program `patest_record`, writing 30sec audio buffers to the disk.

The Software Design

- Need a continuous reading of the audio, and periodic compression.
- After experimentation we created a threaded program, with threads to read audio, file audio, compress data, and other tasks.
- Largely based on the PortAudio test program `patest_record`, writing 30sec audio buffers to the disk.
- File creation, starting compression, and changing output are distributed.

end

Audio Cycle



Compression

- I wanted to use OGG with `oggenc` – takes 80% of elapsed time to compress and gives 1:10 compression
- Alternative is MP3 with `notlame` – takes 74% of elapsed time to compress and gives 1:11 compression
- The program can be built for either – but student DJs think MP3 so I lost for the present....

end

The Other Software

The Other Software

- Small C program renames compressed files by time and moves to daily directory

The Other Software

- Small C program renames compressed files by time and moves to daily directory
- Cronjob deletes directories more than 60 days old

The Other Software

- Small C program renames compressed files by time and moves to daily directory
- Cronjob deletes directories more than 60 days old
- (unfinished) Web-based PERL script to retrieve and deliver timed segments to DJs.

end

Directory for Running System

```
-rw-r--r-- 1 root root 4801096 Mar 7 10:59 Arc0041022.mp3
-rw-r--r-- 1 root root 4801096 Mar 7 11:04 Arc0041023.mp3
-rw-r--r-- 1 root root 4801096 Mar 7 11:09 Arc0041024.mp3
-rw-r--r-- 1 root root 4801096 Mar 7 11:14 Arc0041025.mp3
-rw-r--r-- 1 root root 4801096 Mar 7 11:19 Arc0041026.mp3
-rw-r--r-- 1 root root 4801096 Mar 7 11:24 Arc0041027.mp3
-rw-r--r-- 1 root root 4801096 Mar 7 11:29 Arc0041028.mp3
-rw-r--r-- 1 root root 4801096 Mar 7 11:34 Arc0041029.mp3
-rw-r--r-- 1 root root 4801096 Mar 7 11:39 Arc0041030.mp3
-rw-r--r-- 1 root root 52920000 Mar 7 11:44 Arc0041032
-rw-r--r-- 1 root root 4801096 Mar 7 11:44 Arc0041031.mp3
-rw-r--r-- 1 root root 0 Mar 7 11:47 log5Pw14o
-rw-r--r-- 1 root root 42332160 Mar 7 11:48 Arc0041033
-rw-r--r-- 1 root root 0 Mar 7 11:48 Arc0041034
-rw-r--r-- 1 root root 3145728 Mar 7 11:48 Arc0041032.mp3
```

Experience

Experience

- Been running since October

Experience

- Been running since October
- I forgot about summer time/winter time

Experience

- Been running since Octoberwith no crash.
- I forgot about summer time/winter time

Experience

- Been running since Octoberwith no crash.
- I forgot about summer time/winter timebut we were saved by clock drift of 19sec/day.

Experience

- Been running since Octoberwith no crash.
- I forgot about summer time/winter timebut we were saved by clock drift of 19sec/day.
- Still have problems when the system log is cycled as we lose times for files.

end

Archive Directory

```
-rw-r--r-- 1 root root 4801096 Mar 4 22:55 22:45:08.mp3
-rw-r--r-- 1 root root 4801096 Mar 4 23:00 22:50:08.mp3
-rw-r--r-- 1 root root 4801096 Mar 4 23:05 22:55:08.mp3
-rw-r--r-- 1 root root 4801096 Mar 4 23:10 23:00:08.mp3
-rw-r--r-- 1 root root 4801096 Mar 4 23:15 23:05:08.mp3
-rw-r--r-- 1 root root 4801096 Mar 4 23:20 23:10:07.mp3
-rw-r--r-- 1 root root 4801096 Mar 4 23:25 23:15:07.mp3
-rw-r--r-- 1 root root 4801096 Mar 4 23:30 23:20:07.mp3
-rw-r--r-- 1 root root 4801096 Mar 4 23:35 23:25:07.mp3
-rw-r--r-- 1 root root 4801096 Mar 4 23:40 23:30:07.mp3
-rw-r--r-- 1 root root 4801096 Mar 4 23:45 23:35:07.mp3
-rw-r--r-- 1 root root 4801096 Mar 4 23:50 23:40:07.mp3
-rw-r--r-- 1 root root 4801096 Mar 4 23:55 23:45:07.mp3
-rw-r--r-- 1 root root 4801096 Mar 5 00:00 23:50:07.mp3
-rw-r--r-- 1 root root 4801096 Mar 5 00:05 23:55:07.mp3
-rw-r--r-- 1 root root 6912 Mar 5 01:10 index
```

Conclusions

Conclusions

- Original aims were achieved — a robust logging system for a cost of £60.

Conclusions

- Original aims were achieved — a robust logging system for a cost of £60.
- Still rough edges but usable.

Conclusions

- Original aims were achieved — a robust logging system for a cost of £60.
- Still rough edges but usable.
- The Linux tool-chain and libraries made it much simpler than I feared.

Conclusions

- Original aims were achieved — a robust logging system for a cost of £60.
- Still rough edges but usable.
- The Linux tool-chain and libraries made it much simpler than I feared.
- Still need backups and physical distribution.

end

Conclusions

A fun project, fitting the *what I did in my summer holidays* model. Despite problems a resounding success.

The basic design was dreamt up before and during the warm-up of The Bath University Students' Union T'ai Chi Club. Tom, an official of the club, and I wish to thank our instructor **Simon Lee** for tolerating our improper talking.

