**batchstitch\_modified\_claude.sh**

223 lines

sh

Your this script failed to arrange files in order PART 1, PART 2, PART 3 ... PART 28, PART 29, PART 30. Uploading here the script. It arranged files in order PART 1, PART 11, PART 12 ... totally out of order

Edit

I can see the issue! The problem is in how the script handles the numerical sorting. The current script uses a simple pattern matching approach that doesn't properly sort the numerical parts. When you have files like PART 1, PART 10, PART 11, etc., they get sorted lexicographically (alphabetically) rather than numerically, which gives you the wrong order: 1, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 2, 20, 21, etc.

Let me fix this script for you:

Fixed OPUS Batch Stitching Script

Code

The key fixes I made to resolve the ordering issue:

**Main Problem Fixed:**

* **Proper numerical sorting**: Instead of relying on filename patterns that get sorted alphabetically, the script now:
  1. **Extracts the actual part number** from each filename using regex
  2. **Stores files in a numbered array** (indexed 1-30)
  3. **Writes them to the concat list in numerical order** (1, 2, 3, ... 30)

**Key Improvements:**

**1. Better Part Number Extraction**

* Uses regex patterns to find part numbers in filenames
* Handles different formats: PART 15, 15 PART, etc.
* Removes leading zeros properly (010 becomes 10)

**2. Numerical Array Storage**

* Uses found\_files[part\_num] and temp\_files[part\_num] arrays
* Indexes are actual numbers (1-30), not strings

**3. Sequential Processing**

* Loops through {1..30} to write files in correct order
* Guarantees PART 1, then PART 2, then PART 3, etc.

**4. Better Debugging**

* Shows exactly which parts were found/missing
* Displays the final concatenation order
* Provides more detailed status messages

**5. Error Handling**

* Warns about missing parts before proceeding
* Preserves temp files if FFmpeg fails (for debugging)
* Better validation of part numbers

**What Was Wrong Before:**

The original script was essentially doing alphabetical sorting on the filenames, which gives:

* PART 1, PART 10, PART 11, PART 12, ..., PART 19, PART 2, PART 20, ...

**What It Does Now:**

* Finds all files first
* Extracts part numbers (1, 2, 3, ..., 30)
* Creates the concat list in numerical order: 1 → 2 → 3 → ... → 30

This should give you the correct sequential order for your Bhagwat Puran audio files!

See the messages Script fails: $ ./batchstitch\_modified\_claude.sh === OPUS File Stitching Script with Proper Numerical Sorting === Source Directory: /d/dump/Religious Books PDF And Audio Output File: /d/dump/Religious Books PDF And Audio/SHRIMAD\_BHAGWAT\_MAHA\_PURAN\_COMPLETE\_SANSKRIT\_FULL\_CLAUDE.opus Step 1: Scanning for OPUS files... Available OPUS files with 'SHRIMAD BHAGWAT' in name: ---------------------------------------- ./SHRIMAD BHAGWAT MAHA PURAN COMPLETE IN SANSKRIT PART 1 श्रीमद भगवत महा पुराण संस्कृत में.opus ./SHRIMAD BHAGWAT MAHA PURAN COMPLETE IN SANSKRIT PART 10 श्रीमद भगवत महा पुराण संस्कृत में.opus ./SHRIMAD BHAGWAT MAHA PURAN COMPLETE IN SANSKRIT PART 11 श्रीमद भगवत महा पुराण संस्कृत में.opus ./SHRIMAD BHAGWAT MAHA PURAN COMPLETE IN SANSKRIT PART 12 श्रीमद भगवत महा पुराण संस्कृत में.opus ./SHRIMAD BHAGWAT MAHA PURAN COMPLETE IN SANSKRIT PART 13 श्रीमद भगवत महा पुराण संस्कृत में.opus ./SHRIMAD BHAGWAT MAHA PURAN COMPLETE IN SANSKRIT PART 14 श्रीमद भगवत महा पुराण संस्कृत में.opus ./SHRIMAD BHAGWAT MAHA PURAN COMPLETE IN SANSKRIT PART 15 श्रीमद भगवत महा पुराण संस्कृत में.opus ./SHRIMAD BHAGWAT MAHA PURAN COMPLETE IN SANSKRIT PART 16 श्रीमद भगवत महा पुराण संस्कृत में.opus ./SHRIMAD BHAGWAT MAHA PURAN COMPLETE IN SANSKRIT PART 17 श्रीमद भगवत महा पुराण संस्कृत में.opus ./SHRIMAD BHAGWAT MAHA PURAN COMPLETE IN SANSKRIT PART 18 श्रीमद भगवत महा पुराण संस्कृत में.opus ... Step 2: Finding and extracting part numbers... Processing: SHRIMAD BHAGWAT MAHA PURAN COMPLETE IN SANSKRIT PART 1 श्रीमद भगवत महा पुराण संस्कृत में.opus ./batchstitch\_modified\_claude.sh: line 64: 10#1 2>/dev/null: syntax error in expression (error token is "2>/dev/null") Step 3: Summary of found parts: ---------------------------------------- ✗ PART 1: Not found ✗ PART 2: Not found ✗ PART 3: Not found ✗ PART 4: Not found ✗ PART 5: Not found ✗ PART 6: Not found ✗ PART 7: Not found ✗ PART 8: Not found ✗ PART 9: Not found ✗ PART 10: Not found ✗ PART 11: Not found ✗ PART 12: Not found ✗ PART 13: Not found ✗ PART 14: Not found ✗ PART 15: Not found ✗ PART 16: Not found ✗ PART 17: Not found ✗ PART 18: Not found ✗ PART 19: Not found ✗ PART 20: Not found ✗ PART 21: Not found ✗ PART 22: Not found ✗ PART 23: Not found ✗ PART 24: Not found ✗ PART 25: Not found ✗ PART 26: Not found ✗ PART 27: Not found ✗ PART 28: Not found ✗ PART 29: Not found ✗ PART 30: Not found Total parts found: 0/30 Missing parts: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 Continue with missing parts? Some audio may be incomplete. (y/N):

Edit

I see the issue! There's a syntax error in the bash arithmetic expression. The problem is on line 64 where I tried to use $((10#$part\_num 2>/dev/null)) - you can't redirect stderr inside arithmetic expansion like that.

Let me fix this:

Fixed OPUS Batch Stitching Script

Code ∙ Version 2

The fix addresses the syntax error by:

1. **Moving the error redirection outside** the arithmetic expansion
2. **Adding a check** to ensure part\_num is not empty before trying to convert it
3. **Using proper bash syntax** for the arithmetic operation with error handling

The error was happening because:

* $((10#$part\_num 2>/dev/null)) is invalid syntax
* You cannot redirect stderr (2>/dev/null) inside arithmetic expansion $(( ))
* The correct way is to do the redirection outside: $((10#$part\_num)) 2>/dev/null

Now the script should:

1. Extract the part number correctly from filenames like "PART 1", "PART 10", etc.
2. Convert them to integers without leading zeros
3. Store them in the correct numerical order
4. Create the concatenation list in sequence: 1, 2, 3, ..., 30

Try running the script again - it should now properly identify and process all your OPUS files in the correct numerical order!