EXPLORAREA FORMATULUI PNG

Anton CURMANSCHII,

masterand, anul 2, Facultatea de Matematică și Informatică

CZU: 004.42/.43 anton.curm@gmail.com

Formatul PNG este utilizat pe scară largă în numeroase domenii, incluzând majoritatea aplicațiilor web, aplicații grafice care salvează sau încarcă imagini din fișiere, precum și în dezvoltarea de jocuri. PNG este un format adaptiv și extensibil, având o structură băzată pe chunk-uri, versionate separat unul de altul. Formatul include, de asemenea, mecanisme pentru asigurarea integrității datelor, precum octeții elementului *cyclic redundancy check* (CRC).

Reieșind din toate acestea, formatul PNG merită a fi învățat pentru a manipula eficient fișierele de acest format, pentru a putea extinde formatul prin niște chunk-uri personalizate, și pentru a afla despre metodele de asigurare a integrității, precum și despre algoritmii de compresie folosite ca parte din PNG (Zlib). Aceste idei pot ajuta și la crearea unor formate noi de fișier, nu neapărat de imagine.

Lucrarea de față are scopul educațional să facă cunoștință cu structura formatului PNG [1] și a formatului integrat de compresie Zlib [2-5]. Pentru aceasta s-a decis să se creeze o interfață grafică care să permită explorarea vizuală a formatului PNG.

Conceptul interfeței este următorul: utilizatorul va putea vedea octeții neprocesați din fișier, iar dacă le va accesa, va apărea o fereastră explicativă, informând utilizatorul la ce chunk se referă octetul accesat, și la ce element din acel chunk. De exemplu, în schița interfeței din fig. 1, utilizatorul a accesat pe 80 pe stânga, iar meniul de pe dreapta arată că octetul acesta 80 corespunde elementului *width* și se află în chunk-ul *IHDR,* afișând și alte informații despre chunk.

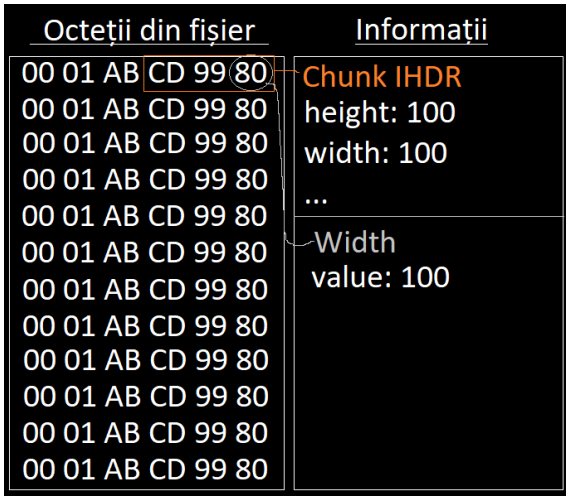


Fig. 1. Schița interfeței grafice

Deoarece programul interacționează cu octeți la nivel scăzut, s-a hotărât să se folosească și un limbaj de programare de nivel scăzut. Limbajul Zig are mai multe instrumente care asigură dacă codul este corect, și este capabil să facă totul ce poate să facă limbajul C, cu toate că nu este un limbaj stabil, fiind în dezvoltare continuă, s-a hotărât să se folosească acest limbaj.

După studierea atentă a specificațiilor PNG și a formatului de compresie integrat Zlib, s-a realizat un parser care înțelege formatele și poate fi folosit pentru a crea un arbore sintactic, reprezentând structura formatului.

Acesta a fost realizat pe baza unei abstracții pentru citire a datelor dintr-un flux de date, inspirată de librăria *System.IO.Pipelines* din .NET [6], care permite să decupleze modul de păstrare a datelor în buferi temporari folosite în procesul de citire încărcate în momentul curent, și propriu zisă citirea din fișier. În principiu, această abstracție permite folosirea programării asincrone sau a unui fir separat pentru a umple buferii cu datele din fișier, însă aceasta nu a fost implementată. Această abstracție joacă rol cheie în arhitectura aplicației (a se vedea fig. 2) — ea permite ca pozițiile elementelor din fișier să fie separate de pozițiile folosite la citire, permițând și ca buferii să se încarce sau descarce din memorie după necesitate. Cu toate că întregul program încă nu este finalizat, abstracția introduce această flexibilitate și acest schelet pentru programare eventuală a sistemului.

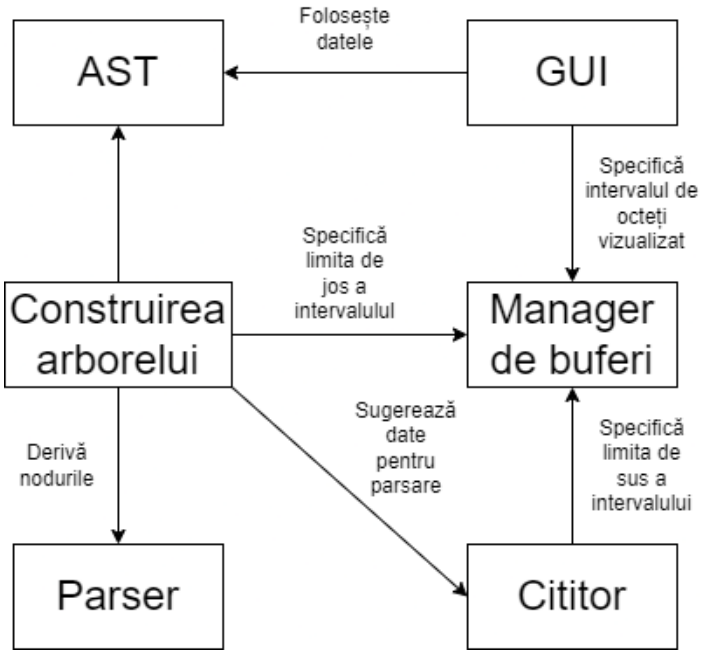


Fig. 2. Arhitectura aplicației

Modulul de construire a arborelui sintactic folosește modulul parserului și gestionează citirea octeților noi din fișier pentru a construi un arbore care poate fi consumat de către modulul de interfață grafică pentru a vizualiza bogat formatul în mod dorit. Fiecare

nod din arbore are o poziție absolută din fișier și formează o structură ierarhică, conform structurii formatului, care permite modulului grafic să lucreze cu această abstracție structurată a arborelui, dar nu direct cu parser-ul. Aceasta simplifică responsabilitățile lui la doar căutarea prin acest arbore.

Totuși, lucrarea prezintă o valoare semnificativă din punct de vedere informativ și educațional pentru autorul însuși, în special în ceea ce privește manipularea fluxurilor de date și construcția unui arbore pe baza acestora, realizarea de abstracții, dezvoltarea interfeței grafice într-un limbaj de programare de nivel scăzut, și, desigur, aprofundarea cunoștințelor despre formatul PNG.

Referințe:

1. PNG (Portable Network Graphics) Specification, Version 1.2. G. Randers-Pehrson et. at 1999. [citat 29 martie 2024]. În: http://www.libpng.org/pub/png/spec/1.2/PNG-Contents.html
2. DEFLATE Compressed Data Format Specification version 1.3. P. Deutsch 1996 [citat 29 martie 2024]. În: https://www.ietf.org/rfc/rfc1951.txt
3. ZLIB Compressed Data Format Specification version 3.3. P. Deutsch 1996 [citat 29 martie 2024]. În: https://datatracker.ietf.org/doc/html/rfc1950
4. How PNG Works: Compromising Speed for Quality. Reducible 2022 [citat 29 martie 2024]. În: https://www.youtube.com/watch?v=EFUYNoFRHQI
5. Huffman Codes: An Information Theory Perspective. Reducible 2021 [citat 29 martie 2024]. În: https://youtu.be/B3y0RsVCyrw
6. System.IO.Pipelinesin.NET.Microsoft2022.[citat29martie2024].În:https://learn.microsoft.com/en-us/dotnet/standard/io/pipelines

Recomandat
***Sergiu PERETEATCU**, dr., conf. univ.*