**Universiteti i Prishtinës “Hasan Prishtina”**

**Fakulteti Inxhinierisë Elektrike dhe Kompjuterike**

****

**Dokumentim teknik i projektit**

**Lënda: Sisteme Operative**

**Titulli i projektit: Priority Scheduling**

**Emri profesorit/Asistentit Emri & mbiemri studentëve / email adresa**

|  |  |  |
| --- | --- | --- |
| Prof. Dr. Artan Mazrekaj  Ass. Dalina Vranovci | 1. Donat Sinani | donat.sinani@student.uni-pr.edu |
| 2. Enes Hasani | enes.hasani@student.uni-pr.edu |
| 3. Erlis Lushtaku | erlis.lushtaku@student.uni-pr.edu |

Prishtinë, 2021

Përmbajtja

[**Abstrakti** 3](#_Toc94128872)

[**I.** **Hyrje** 4](#_Toc94128873)

[**II.** **Qëllimi i punimit** 5](#_Toc94128874)

[**III.** **Pwrshkrimi i problemit** 5](#_Toc94128875)

[**IV.** **Teknologjia** 6](#_Toc94128876)

[**V.** **Implementimi** 6](#_Toc94128877)

[**VI.** **Union Based Sql Injection *[5][6]*** 14](#_Toc94128878)

[1. Përcaktimi i numrit të kolonave të kërkuara në një sulm UNION based SQLi 14](#_Toc94128879)

[2. Gjetja e kolonave me një tip të dhënash të përshtatshëm për sulmin UNION based SQLi 15](#_Toc94128880)

[3. Përdorimi i sulmit UNION based SQLi për marrjen e të dhënave që na interesojnë 15](#_Toc94128881)

[4. Skripta UnionSqli.py 16](#_Toc94128882)

[5. Ekzekutimi i sulmit duke përdorur Burp Suite Proffesional 18](#_Toc94128883)

[6. Skripta Requests.py *[7]* 30](#_Toc94128884)

[**VII.** **Konkluzione** 32](#_Toc94128885)

[**Referencat** 33](#_Toc94128886)

# **Abstrakti**

Njesia qendrore procesorike(CPU) konsiderohet burimi kryesor dhe më i rëndësishëm në sistemin kompjuterik. Skedulimi i CPU-së wshtw procedura e zgjedhjes sw njwrit nga proceset qw gjenden nw rreshtin e proceseve tw gatshme(ready queue) tw cilit do t’i alokohet CPU-ja, proceset e tjera presin radhwn e tyre pwr alokim tw CPU-sw nw ready queue. Pasi qw CPU-ja wshtw burimi mw i rwndsishwm nw sistemin kompjuterik dhe duhet tw shfrytwzohet me kujdes, duke kwmbyer(switch) CPU-nw midis proceseve, sistemi operativ mund të bëjë kompjuterin më produktiv. Andaj algoritmet e skedulimit qw zgjidhin problemin e skedulimit bwhen shumw tw rwndwsishme nw arritjen e qwllimeve tw dizajnimit tw njw sistemi operativ(OS) qw nw mwnyrw efikase menaxhon resurset harduerike tw sistemit kompjuterik pwrkatwsisht CPU-nw. Ekzistojnw disa algoritme tw skedulimit, ndwr mw tw rwndwsishmet janw: First-Come First-Serve(i pari vjen, i pari sherbehet) Scheduling, Shortest-Job-First(puna mw e shkurtw nw fillim) Scheduling, Round-Robin Scheduling, Priority(prioritet) Scheduling dhe Multilevel Queue(rreshti disa nivelwsh) Scheduling. Secili nga kwto algoritme ka pwrparwsitw dhe dobesitw e veta pwrgjatw pwrdorimit, andaj varwsisht nga problemi zgjedhet njwri nga algoritmet. Ne pwrgjatw kwtij punimi do tw zjidhim problemin e skedulimit duke e pwrdorur algoritmin priority-schedueling.

**Fjalwt kyçe:** Njesia qendrore procesorike(CPU),Skedulimi i CPU-së, ready queue, algoritmet e skedulimit**,** First-Come, First-Serve Scheduling, Shortest-Job-First Scheduling, Round-Robin Scheduling, Priority Scheduling, Multilevel Queue Scheduling

# **Hyrje**

Algoritmi priority-scheduling secilit proces i asocon një prioritet të caktuar, dhe e alokon CPU-nw për procesin me prioritetin më tw lartw. Prioriteti mund tw jetw i pwrcaktuar nw mwnyra tw ndryshme nw sisteme tw ndryshme, vlera mw e madhe mund tw ketw prioritet mw tw lartw, ose e kunwrta. Njw lloj i veçantw i priority-scheduling wshtw SJF(Shortest Job First) ku proceset tw cilat burst time(kohwn e ekzekutimit) e kanw mw tw vogwl kanw prioritet mw tw lartw. Nw rast se dy procese me prioritete identike gjenden nw ready queue, atwherw ato skedulohen me FCFS(First Come First Serve).

Kemi dy lloje tw algoritmit priority-scheduling:

* Preemptive priority-scheduling

Nw rast se njw proces ka alokuar CPU-nw, dhe wshtw duke u ekzekutuar, nw kohwn e njejtw vjen nw ready queue njw proces i ri me prioritet mw tw lartw, atwherw procesi qw wshtw duke u ekzekutuar liron CPU-nw, dhe procesi i ri alokon CPU-nw. Procesi i cili wshtw bllokuar vazhdon punwn e tij(alokon prapw CPU-nw) vetwm nwse nuk ka prapw njw proces tw ri nw ready queue me prioritet mw tw lartw.

* Non-preemptive priority-scheduling

Procesi i cili e alokon CPU-nw nuk e liron atw deri sa tw pwrfundojw punwn e tij, pa marw parasysh a gjendet ndonjw proces nw ready queue me prioritet mw tw lartw.

Njw problem i cili haset te ky algoritwm wshtw starvation, i cili ka kuptimin qw njw proces me prioritet tw ulwt mund tw mos ekzekutohet pwr njw kohw tw gjatw, pra ai wshtw i gatshwm pwr tu ekzekutuar, gjendet nw ready queue, por nuk e alokon CPU-nw ngase proceset tjera me prioritet mw tw lartw e alokojnw atw.

Njw zgjidhje wshtw aging, e ka kuptimin qw njw proces pas qwndrimit tw njw kohe tw caktuar nw ready queue, i rritet prioriteti nw mwnyrw periodike (pwr shembull qdo sekond i rritet prioriteti pwr 1) qw tw mund tw alokojw CPU-nw.

# **Qëllimi i punimit**

Qwllimi i kwtij punimit wshtw tw demostrojw si funksionon algoritmi Priority Scheduling, pwrkatwsisht Preemptive Priority-Scheduling, algoritwm i cili edhe do tw kodohet.

**Avantazhet e Priority Scheduling:**

* Proceseve kritike u asociohet prioritet mw i lartw
* Proceset me prioritet tw lartw ekzekutohen mw pwrpara(e rwndwsishme pwr sisteme me kohw reale-real time systems)
* Rwndwsia e secilit proces wshtw e pwrcaktuar me precizitet

**Disavantazhet e Priority Scheduling:**

* Nwse nw vazhdimsi ndodh qw proceset me prioritet tw lartw kwrkojnw CPU-nw atwherw ato me prioritet tw ulwt mund t’iu duhet tw presin pwr njw kohw pafundwsisht tw gjatw
* Nwse sistemit i ndodh ndonjw crash mund t’i humbim tw gjitha proceset me prioritet tw ulwt
* Nwse procesi me prioritet tw lartw zgjatw shumw po ashtu mund tw ndodh starvation(procesi me prioritet tw ulwt nuk ekzekutohet).

# **Pwrshkrimi i problemit**

Nw kwtw punim problemi qw do tw trajtohet wshtw skedulimi i njw numri tw çfardoshwm tw proceseve ku si informata rreth proceseve qw do tw skedulohen kemi:

* kohwzgjatjen e procesit(burst time) – njw numwr qw tregon kohwn qw i duhet procesit ta mbajw nwn shwrbim CPU-nw
* prioritetin e procesit - njw numwr qw tregon rwndwsinw e procesit, numri mw i vogwl rwndwsia mw e madhe dhe anasjelltas
* kohwn e arritjes(arrival time) – koha kur procesi mbwrrinw nw ready queue dhe wshtw i gatshwm pwr ekzekutim, gjendje e njohur si gjendje gatishmwrie(ready state).

Skedulimin do ta bwjmw sipas algoritmit Priority Scheduling, pwrkatwsisht Preemptive Priority-Scheduling.

Pasi qw algoritmi i skedulimit ta pwrfundojw skedulimin e proceseve ne do tw shohim:

* kohwzgjatja e ekzekutimit(burst time) – koha totale qw i duhet njw procesi tw ekzekutohet(ta mbajw nwn shwrbim CPU-nw)
* koha e pritjes (waiting time) – koha totale qw kalon procesi nw gjendje gatishmwrie(ready state/queue) duke pritur tw marr nwn shwrbim CPU-nw
* koha e pwrgjigjjes(response time) – koha qw kalon qw nga momenti kur procesi gjendet nw gjendje gatishmwrie(ready state) e deri sa ta merr nwn shwrbim CPU-nw pwr herw tw parw
* koha totale e pwrfundimit(turnaround time) – wshtw koha totale qw kalon procesi qw nga ardhja nw ready queue pwr herw tw parw e deri nw pwrfundim

# **Teknologjia**

Pwr zhvillimin e kwtij programi qw implementon algoritmin priority scheduling wshtw pwrdorur gjuha programuese C++, kodi wshtw shkruajtur nw editorin Visual Studio, ndwrsa sistemi operativ i pwrdorur wshtw Windows.

# **Implementimi**

## Funksioni fillVector

## Funksioni orderVector

Ky funksion merr si parameter

## Funksioni printVector

# **Konkluzione**

# **Referencat**

|  |  |
| --- | --- |
| [1] | <https://sectigostore.com/blog/42-cyber-attack-statistics-by-year-a-look-at-the-last-decade/> |
| [2] | <https://heimdalsecurity.com/blog/10-surprising-cyber-security-facts-that-may-affect-your-online-safety/> |
| [3] | <https://null-byte.wonderhowto.com/how-to/discover-xss-security-flaws-by-fuzzing-with-burp-suite-wfuzz-xsstrike-0189971/> |
| [4] | <https://null-byte.wonderhowto.com/how-to/hack-like-pro-hack-web-apps-part-4-hacking-form-authentication-with-burp-suite-0163642/> |
| [5] | <https://portswigger.net/web-security/sql-injection/union-attacks> |
| [6] | https://www.youtube.com/watch?v=n2YxdgX5SJA&list=PLuyTk2\_mYISLaZC4fVqDuW\_hOk0dd5rlf&index=11 |
| [7] | <https://github.com/rkhal101/Web-Security-Academy-Series/blob/main/sql-injection/lab-09/sqli-lab-09.py> |