**KONSPEKT PRZEDMIOTU**

Semestr letni, rok akad. 2020/2021

|  |  |
| --- | --- |
| Nazwa przedmiotu | Systemy operacyjne (Linux) |
| Kierunek/-i studiów / rok studiów / semestr studiów | Informatyka/2/IV |
| Tryb studiów | Niestacjonarny |
| Forma zajęć | Laboratorium |
| Liczba godzin | 16 |
| Koordynator przedmiotu: | mgr inż. Łukasz Laszko |
| Jakie są ogólne cele dydaktyczne tego przedmiotu? | 1. Przyswojenie wiedzy w zakresie podstawowych mechanizmów i zasad działania współczesnych systemów operacyjnych.  2. Nabycie podstawowych umiejętności w zakresie korzystania z systemu operacyjnego Linux.  3. Nabycie podstawowych umiejętności w zakresie programowania systemowego w systemie operacyjnym Linux. |
| Jak są kryteria zaliczenia tego przedmiotu? | Za ćwiczenia laboratoryjne studenci mogą uzyskać 40 punktów. Punkty student zdobywa po realizacji wydanych zadań na zajęciach, zadań domowych oraz na sprawdzianie. Laboratoria uznaje się za zaliczone jeżeli student uzyska łącznie za wszystkie ćwiczenia co najmniej 16 punktów, co stanowi 40% osiągniętych efektów uczenia się. Za ćwiczenia laboratoryjne wystawiana jest ocena uogólniona (zal/nzal). |
| Jakie są kryteria zaliczenia tego przedmiotu na ocenę celującą? | Uzyskanie z laboratoriów co najmniej 85% z 40 możliwych punktów i realizacja wszystkich dodatkowych zadań laboratoryjnych oraz uzyskanie na egzaminie co najmniej 85% z 60 możliwych punktów. |
| Pozostałe informacje, dotyczące tego przedmiotu, ważne dla studenta | - Za aktywności i prace domowe można uzyskać maks. 25 pkt.  - Za sprawdzian można uzyskać maks. 15 pkt.  Sprawdzian nie ma progu zaliczenia.  Student, który nie był obecny na co najmniej 15% zajęć w semestrze nie może przystąpić do zaliczenia poprawkowego. |

|  |  |
| --- | --- |
| **ZAJĘCIA 1**  **(2 godz.)**  **Środowisko systemu Linux (wprowadzenie)** | |
| Po tych zajęciach student (będzie znał/ wiedział/ umiał/ potrafił / rozumiał) | Efekty uczenia się:   1. Student będzie znał zasady korzystania z trybu graficznego oraz tekstowego systemu Linux. 2. Student będzie umiał korzystać z systemu pomocy. 3. Student będzie umiał posługiwać się podstawowymi poleceniami powłoki systemowej w zakresie przechodzenia pomiędzy katalogami na dysku i listowania zawartości dysku. |
| Treści zajęć | 1. Ogólne wprowadzenie w środowisko systemu Linux. 2. System pomocy – man, info, apropos, „--help”, „-v”. 3. System plików – ls, dir, pwd, cd. 4. Struktura systemu plików i przeznaczenie najważniejszych katalogów. 5. Typy plików – ls, test, file. |
| Źródło/a do nauki treści z tych zajęć | Minimalne / obowiązkowe:  - Podręcznik systemowy (man, info) dla omawianych poleceń i programów systemowych.  - Sosna Ł., Linux - komendy i polecenia, 2010  Rozszerzające / uzupełniające:  - Garrels M., Bash Guide for Beginners, 2008 |

|  |  |
| --- | --- |
| **ZAJĘCIA 2**  **(2 godz.)**  **Środowisko systemu Linux (wybrane polecenia i programy użytkowe)** | |
| Po tych zajęciach student (będzie znał/ wiedział/ umiał/ potrafił / rozumiał) | Efekty uczenia się:   1. Student będzie umiał wykonywać podstawowe operacje na plikach i katalogach. 2. Student będzie znał prawa dostępu do plików i katalogów i będzie umiał zmieć te prawa. 3. Student będzie umiał utworzyć plik, wyedytować go i zapisać. |
| Treści zajęć | 1. Operacje na plikach/katalogach – touch, cp, mv, rm, mkdir, rmdir, cat, less. 2. Prawa dostępu – chmod, chown, chgrp, umask. 3. Inne przydatne polecenia i programy – vi, echo, uname, date, who, id, df, du, su, sudo, ssh, scp, passwd, tar. |
| Źródło/a do nauki treści z tych zajęć | Minimalne / obowiązkowe:  - Podręcznik systemowy (man, info) dla omawianych poleceń i programów systemowych.  - Sosna Ł., Linux - komendy i polecenia, 2010  Rozszerzające / uzupełniające:  - Garrels M., Bash Guide for Beginners, 2008 |

|  |  |
| --- | --- |
| **ZAJĘCIA 3**  **(2 godz.)**  **Środowisko systemu Linux (strumienie, potoki, sterowanie procesami)** | |
| Po tych zajęciach student (będzie znał/ wiedział/ umiał/ potrafił / rozumiał) | Efekty uczenia się:   1. Student będzie rozumiał ideę komunikacji między procesami w powłoce systemowej. 2. Student będzie umiał sterować strumieniami procesów. 3. Student będzie umiał sterować procesami i zadaniami w powłoce systemowej. 4. Student będzie umiał korzystać z programów do monitorowania zasobów systemu. |
| Treści zajęć | 1. Przeadresowanie strumieni, przetwarzanie potokowe - operatory >, >>, <, <<, |, polecenie xargs. 2. Sterowanie procesami – ps, pstree, kill, killall, sleep. 3. Sterowanie zadaniami przez używaną powłokę (procesy pierwszo- i drugoplanowe), wstrzymywanie i usuwanie procesów - fg, bg, jobs. kill, killall, pidof, sygnały ^Z i ^C. 4. Monitorowanie procesów i wykorzystania zasobów systemu – top, htop, free, uptime, df, du. |
| Źródło/a do nauki treści z tych zajęć | Minimalne / obowiązkowe:  - Podręcznik systemowy (man, info) dla omawianych poleceń i programów systemowych.  - Sosna Ł., Linux - komendy i polecenia, 2010  Rozszerzające / uzupełniające:  - Garrels M., Bash Guide for Beginners, 2008 |

|  |  |
| --- | --- |
| **ZAJĘCIA 4**  **(2 godz.)**  **Środowisko systemu Linux (wprowadzenie do skryptów powłoki).**  **Programowanie w języku powłoki (przekazywanie parametrów do skryptów)** | |
| Po tych zajęciach student (będzie znał/ wiedział/ umiał/ potrafił / rozumiał) | Efekty uczenia się:   1. Student będzie umiał: filtrować, sortować, przeszukiwać oraz przetwarzać tekst w powłoce systemowej. 2. Student będzie umiał korzystać ze zmiennych w powłoce systemowej. 3. Student będzie umiał napisać i uruchomić prosty skrypt powłoki. |
| Treści zajęć | 1. Filtrowanie odpowiedzi aplikacji użytkowych – tr, cut. 2. Sortowanie tekstu – sort, uniq. 3. Wyrażenia regularne – grep. 4. Operacje na tekście – sed, awk. 5. Zmienne i ich typy – env, set, unset, export. 6. Cytowanie poleceń – eval, operator „`”. 7. Proste skrypty powłoki i parametry przekazywane do skryptów. |
| Źródło/a do nauki treści z tych zajęć | Minimalne / obowiązkowe:  - Podręcznik systemowy (man, info) dla omawianych poleceń i programów systemowych.  - Sosna Ł., Linux - komendy i polecenia, 2010  Rozszerzające / uzupełniające:  - Garrels M., Bash Guide for Beginners, 2008 |

|  |  |
| --- | --- |
| **ZAJĘCIA 5**  **(2 godz.)**  **Programowanie w języku powłoki (funkcje, instrukcje sterujące, operacje arytmetyczne)** | |
| Po tych zajęciach student (będzie znał/ wiedział/ umiał/ potrafił / rozumiał) | Efekty uczenia się:   1. Student będzie umiał wykorzystywać funkcje wewnątrz skryptów powłoki. 2. Student będzie umiał posługiwać się instrukcjami sterującymi, pętlami oraz zmiennymi tablicowymi w swoich skryptach powłoki. 3. Student będzie umiał wykonywać operacje arytmetyczne w powłoce. |
| Treści zajęć | 1. Funkcje w skryptach powłoki. 2. Instrukcje sterujące – if-elif-else, case. 3. Pętle – for, while, until. 4. Zmienne tablicowe. 5. Operacje arytmetyczne – let, expr, bc. |
| Źródło/a do nauki treści z tych zajęć | Minimalne / obowiązkowe:  - Podręcznik systemowy (man, info) dla omawianych poleceń i programów systemowych.  - Sosna Ł., Linux - komendy i polecenia, 2010  Rozszerzające / uzupełniające:  - Garrels M., Bash Guide for Beginners, 2008  - Math Arithmetic: How To Do Calculation In Bash? https://www.shell-tips.com/bash/math-arithmetic-calculation/ |

|  |  |
| --- | --- |
| **ZAJĘCIA 6**  **(2 godz.)**  **Procesy i synchronizacja procesów (programowanie systemowe w C)** | |
| Po tych zajęciach student (będzie znał/ wiedział/ umiał/ potrafił / rozumiał) | Efekty uczenia się:   1. Student będzie umiał napisać prosty program w języku C. 2. Student będzie umiał programistycznie powoływać nowe procesy w swoim programie, zgodnie z podaną przez prowadzącego specyfikacją. 3. Student będzie umiał sterować procesami z poziomu kodu programu. |
| Treści zajęć | 1. Powoływanie procesów potomnych – fork(). 2. Tworzenie rodziny procesów – wait(), waitpid(). |
| Źródło/a do nauki treści z tych zajęć | Minimalne / obowiązkowe:  - Podręcznik systemowy (man, info) dla omawianych funkcji języka C.  - Love R., LINUX - programowanie systemowe, 2014  Rozszerzające / uzupełniające:  - fork() in C, https://www.geeksforgeeks.org/fork-system-call/  - Stevens W.R.. UNIX programowanie usług sieciowych, Tom 2, 2001 |

|  |  |
| --- | --- |
| **ZAJĘCIA 7**  **(2 godz.)**  **Procesy i synchronizacja procesów (programowanie systemowe w C).**  **Komunikacja międzyprocesowa (programowanie systemowe w C)** | |
| Po tych zajęciach student (będzie znał/ wiedział/ umiał/ potrafił / rozumiał) | Efekty uczenia się:   1. Student będzie umiał wykorzystywać wielowątkowość w swoim programie. 2. Student będzie umiał programistycznie korzystać z poleceń i programów powłoki. 3. Student będzie umiał programistycznie wykorzystać potoki (pipe) do skomunikowania procesów. |
| Treści zajęć | 1. Powoływanie procesów lekkich (wątki POSIX) – pthread\_create(), pthread\_join(), pthread\_exit(). 2. Uruchamianie poleceń systemowych i innych programów – system(), execl(). 3. Łącza nienazwane (pipe’y) – pipe(), close(). |
| Źródło/a do nauki treści z tych zajęć | Minimalne / obowiązkowe:  - Podręcznik systemowy (man, info) dla omawianych funkcji języka C.  - Love R., LINUX - programowanie systemowe, 2014  Rozszerzające / uzupełniające:  - Multithreading in C, https://www.geeksforgeeks.org/multithreading-c-2/  - Stevens W.R.. UNIX programowanie usług sieciowych, Tom 2, 2001 |

|  |  |
| --- | --- |
| **ZAJĘCIA 8**  **(2 godz.)**  **Komunikacja międzyprocesowa (programowanie systemowe w C).**  **Repetytorium i sprawdzian** | |
| Po tych zajęciach student (będzie znał/ wiedział/ umiał/ potrafił / rozumiał) | Efekty uczenia się:   1. Student będzie umiał programistycznie wykorzystać kolejki fifo (named pipe) do skomunikowania procesów. 2. Student będzie umiał obsłużyć sygnał (komunikat systemowy, przerwanie programowe) w swoim programie. 3. Student będzie potrafił wykonać prostą komunikację asynchroniczną pomiędzy procesami w swoim programie. 4. Student będzie miał możliwość uzupełnienia swojej wiedzy i umiejętności w zakresie zrealizowanych w semestrze zajęć laboratoryjnych. |
| Treści zajęć | 1. Łącza nazwane (kolejki FIFO) – mkfifo(), mknod(), unlink(), polecenie mkfifo. 2. Sygnały i ich obsługa – signal(), kill(), polecenie trap. 3. Repetytorium. 4. Sprawdzian testowy z zakresu tematów zrealizowanych podczas laboratorów 1-5. |
| Źródło/a do nauki treści z tych zajęć | Minimalne / obowiązkowe:  - Podręcznik systemowy (man, info) dla omawianych funkcji języka C.  - Love R., LINUX - programowanie systemowe, 2014  Rozszerzające / uzupełniające:  - Signals in C language, https://www.geeksforgeeks.org/signals-c-language/  - Stevens W.R.. UNIX programowanie usług sieciowych, Tom 2, 2001 |