



WYKŁADY

1. Wstęp

- System operacyjny definicja
 - Warstwa oprogramowania, która kontroluje dostęp programów użytkowych do **zasobów komputera** i umożliwia **komunikację** między nimi
- pakiety
 - rpm - fedora
 - deb - debian

2. Shell

- co to shell
 - **interfejs użytkownika** dostępu do usług i zasobów os
- logowanie i zmiana hasła
 - passwd
 - exit
- opcje
 - -n krótkie opcje
 - --number długie opcje
- kim jestem?
 - **whoami** - nazwa użytkownika
 - who am i - whoami.+ dodatkowe info
 - who - zalogowani uzytkownicy
- procesy
 - **ps**
 - **ps -ef**
 - **ps ax**
 - zatrzymanie procesu **ctrl+Z (Zamrożenie)**
 - przerwanie procesu **ctrl+C**
 - procesy w tle **nazwa_procesu &**
 - **jobs** - wyświetlenie zadań
 - **bg/fg [id]** - odtrzymanie procesu itd
 - kill -l - lista sygnałów
 - **kill [-numer_sygnału] [PID]**
 - **kill wysyła sygnał do procesu**
- katalogi (WAŻNE!)
 - / - root
 - **/bin/** - podstawowe programy systemowe
 - **/dev/** - pliki urządzeń (od devices)
 - **/etc/** - configi
 - **/home/** - katalogi domowe
 - **/lib/** - podstawowe biblioteki
 - **/media/** - montowanie urządzeń
 - **/mnt/** - montowanie partycji
 - **/proc/** - pliki dostępne do OS
 - **/root/** - katalog domowy roota
 - **/sbin/** - podstawowe programy administracyjne
 - **/sys/** - pliki dostępne do OS
 - **/tmp/** - pliki tymczasowe



SO EGZAMIN NOTATKA

- **/usr/** - pliki wykonywalne, biblioteki, programy programów użytkowych
 - **/usr/bin/** - pliki wykonalne programów użytkowych
 - **/usr/lib/** - biblioteki programów użytkowych
 - **/usr/share/** - dodatkowe pliki programów użytkowych
- **/var/** - pliki dynamiczne
- obecny katalog - **pwd**
- strumienie
 - stdin. program < input.txt
 - stdout. program > output.txt
 - stderr. program 2> error.txt
 - | - pipe
 - program >> output.txt (append)
- **head, tail** - początek, koniec pliku -nr_linii
- **less** - przeglądanie dokumentów
- **wc** - statystyki pliku (word count)
- **grep** [OPCJE] WZORZEC [PLIK]
 - -i = ignore case
 - -v = bez wzorca
 - -n = numery linii
 - -c = policzenie matchy
- **find** [OPCJE] [ścieżka ...]
 - **-name** [wzorzec]
 - **-type** f/d
- oddzielanie poleceń
 - **sekwencja ; sekwencja2**
 - **to && to2 (warunek to2, to się wykonało)**
 - **to || to2 (warunek to2, to się nie wykonało)**
- \$PATH - ścieżka, w której Bash poszukuje pliki do wykonania
- zmienne specjalne
 - **\$0** nazwa skryptu
 - **\$1...9** parametry kolejne
 - **\$#** argc
 - **"\$"** wszystkie parametry
 - **\$\$** - obecne pid
- 3. Skrypty
 - prawa dostępu
 - **ugo = drwxrwxrwx**
 - user group others
 - chmod ugo np **chmod 755 file.sh**
 - **chmod +x file** nadawanie uprawnień
 - chmod u-x file odbieranie uprawnień
 - chmod u=rwx ustalanie uprawnień
 - chmod [ugo][+/-/=][rwx]
 - **#!/bin/sh**
 - tablice
 - TAB=(a b c d)
 - **declare -A ASOCJACYJNA** stworzenie tablicy asocjacyjnej
 - ASOCJACYJNA[klucz] = wartość
 - zawartość **\${TABLICA[klucz]}**
 - cała zawartość **\${TABLICA[*]}**

SO EGZAMIN NOTATKA

- funkcja() {; }
- parametry funkcji \$1..\$n
- **instrukcje sterujące**
 - if [[warunek]]; then <polecenia>; fi
 - warunek [[TEST]]
 - [[TEST1 && TEST2]] koniunkcja
 - [[TEST1 || TEST2]] alternatywa
 - [[\$X == \$Y]]
 - [[-e PLIK]] plik istnieje
 - [[-f PLIK]] plik istnieje i jest f
 - [[-d PLIK]] plik istnieje i jest d
 - [[-L PLIK]] plik jest symlinkiem
 - [[-r/w/x PLIK]] plik istnieje i ma r w lub x
 - case <wyrażenie> in
 - <label1> traktatata ;;
 - <label2> traktatata;;
 - *) default;;
 - esac
 - while [[warunek]]; do <zbiór instrukcji>; done
 - for X in a b c; do <instrukcje>; done
 - for ((i = 0; i < 10; i++)); do <instrukcje>; done
 - **inkrementacja i=\$((i+1))**
- pliki tymczasowe
 - **tmpfs** (w ramie) (katalog /tmp)
 - TEMP=\$(mktemp)
 - #komentarz
- przetwarzanie tekstu
 - sort
 - -k numer kolumna
 - -r odwrócona kolejność
 - -n
 - uniq
 - -c policzenie
 - wc
 - cut
 - -d “,” delimiter
 - -f 1,2,3 kolumny
 - diff
 - tr [tekst] [tekst na który zamienić]
 - sed
 - awk
- 4. Administracja systemów
 - dependency
 - łańcuch zależności
 - konflikt zależności
 - circular dependency
 - pakiety
 - rodzaje pakietów
 - deb - Debian
 - rpm - fedora
 - tgz - Slackware

SO EGZAMIN NOTATKA

- portage - Gentoo
- pkgsrc - BSD
- systemy zarządzania pakietami
 - dpkg
 - pacman
 - apt
- serwery www
 - udostępnianie danych
 - Apache
 - PostgreSQL
 - LAMPP (Linux Apache MySql PHP/Python/Perl)
- MTA - serwer poczty
 - postfix
- serwer plików SAMBA
- sieć
 - ip addr
 - ifconfig
 - netstat
 - firewall
 - packet filtering firewall
 - filtr kontekstowy
 - proxy
 - cachowanie
 - sieć chroniona
- **SELinux**
 - Security-Enhanced Linux
 - MAC - mandatory access control
 - RBAC - role-based access control
- **IDS** - system wykrywania włamań
 - monitorowanie ruchu sieciowego
 - NIDS - network intrusion detection system
 - HIDS - host intrusion detection system
 - analiza znanych ataków
 - anomalie ruchu sieciowego
 - IDS + firewall = IPS
- SIEM
 - np. Snort
 - kolekcja danych
 - logowanie
 - dashboard
- Systemy ochrony integralności plików

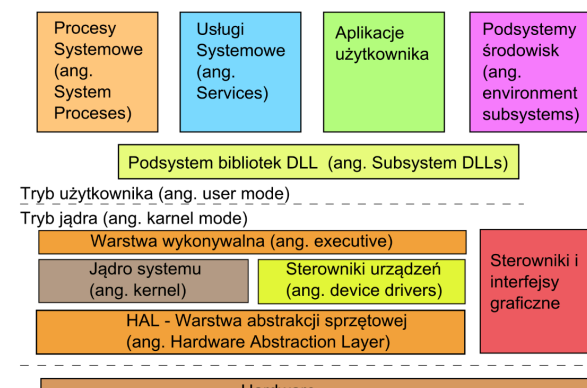
- **skrót kryptograficzny**

- wybranie plików do ochrony
- lista zmienionych plików
- baza skrótów
- np. tripwire

5. Windows

- architektura windowsa
 - **tryb użytkownika**
 - procesy systemowe np. logowanie, menedżer sesji

Architektura systemu MS Windows



SO EGZAMIN NOTATKA

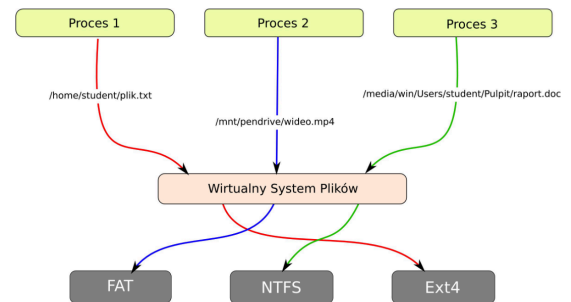
- usługi systemowe np.harmonogram zadań
- aplikacje użytkownika
- podsystem dll
 - **winapi**
 - udokumentowane
- **tryb jądra**
 - warstwa wykonywalna (executive)
 - zarządzanie pamięcią, procesami, wątkami
 - obsługa sieci
 - I/O
 - jądro systemu
 - szeregowanie procesów, obsługa przerwań, wyjątków
 - synchronizacja procesów
 - funkcje wykorzystywane przez wyższe warstwy
 - sterowniki
 - sprzętu
 - system plików
 - **HAL** - warstwa abstrakcji sprzętowej
 - niezależność od różnic sprzętowych
 - sterowniki i interfejsy graficzne
 - GUI
 - sterowniki kart graficznych
 - rysowanie, okienka itd.
- logowanie do systemu
 - login
 - .\login
 - NAZWA_KOMPUTERA\login
 - NAZWA_DOMENY\login
- polecenia cmd
 - dir, cd, help
 - wmic
 - net
 - ipconfig
 - netstat
 - pingowanie
 - ping
 - pathping
 - tracert
- narzędzia administracyjne
 - włączanie: panel sterowania-> narzędzia administracyjne
 - konfiguracja systemu
 - zdarzenia
 - Usługi
- rejestr
 - edycja: regedit
 - struktura hierarchiczna
 - informacje o:
 - sprzęcie
 - jądrze systemu
 - usługach
 - UI

SO EGZAMIN NOTATKA

- aplikacjach
- klucze: katalogi
- wartości: pliki
- struktura rejestru
 - **HKCR** (HKEY_CLASSES_ROOT)
 - powiązania plików i aplikacji
 - **HKCU** (HKEY_CURRENT_USER)
 - dane obecnego użytkownika
 - AppEvents - zdarzenia
 - Console - config cmd
 - Control Panel - pulpit, klawiatura
 - Environment - zmienne środowiskowe
 - Keyboard Layout
 - Network
 - Printers
 - Software - konfiguracja aplikacji użytkownika
 - **HKLM** (HKEY_LOCAL_MACHINE)
 - dane systemu operacyjnego
 - **HKU** (HKEY_USERS)
 - dane dot. wszystkich użytkowników
 - może wystąpić HKCC (HKEY_CURRENT_CONFIG)
 - dostępny sprzęt
 - może HKEY_PERFORMANCE_DATA
 - wydajność
 - może HKEY_DYN_DATA
 - dane o urządzeniach
- typy danych w rejestrze eg.
 - REG_NONE brak typu
 - REG_SZ ciąg znaków stałej długości
 - REG_EXPAND_SZ ciąg znaków zmiennej długości
 - REG_BINARY binarne
 - REG_DWORD liczba 32 bitowa
 - REG_LINK odsyłacz
 - REG_MULTI_SZ ciąg znaków zakończony NULL
 - REG_RESOURCE_LIST zasoby sprzętu
 - REG_QWORD liczba 64-bitowa
- **winapi**
 - windows.h
 - notacja węgierska
 - LPSTR - string*
 - UINT
 - HWND - window handle
 - TCHAR - znak unicode lub nie
 - _T("text") konwersja do unicode
 - obsługa komunikatów
- **windows server**
 - asp (active server pages)
 - .NET, C#, C++
 - PHP
 - serwer www, ftp, dhcp
 - bazy danych

SO EGZAMIN NOTATKA

- azure
 - tworzenie aplikacji w chmurze
 - maszyny wirtualne
- 6. Systemy plików
 - system plików - reguły zapisu danych na nośniku
 - cele stosowania systemów plików
 - przechowywanie danych
 - szybki dostęp do danych
 - kontrola błędów
 - dostarczanie informacji o zapisanych plikach
 - **VFS** (virtual file system)
 - pośredniczy między rzeczywistym systemem plików a funkcjami użytkownika
 - interfejs programistyczny systemu plików niezależny od rzeczywistego systemu plików
 - przykładowe systemy plików
 - Linux
 - Minix
 - Ext, Ext2, Ext3, **Ext4** (Extended file system)
 - Btrfs
 - Windows
 - FAT16, FAT32 (File Allocation Table)
 - **NTFS** (New Technology File system)
 - Mac os
 - HFS+
 - BSD
 - ZFS
 - **bloki**
 - lista blokowa O.O
 - redukcja liczby odczytów i zapisów
 - FAT16, FAT32
 - plik zapisywany w blokach
 - brak znaku EOF ale jest indeks następnego bloku
 - ośmioznakowe nazwy (FAT16), długie nazwy (FAT32)
 - **NTFS**
 - prawa dostępu do plików
 - metadane plików
 - **rozdzielczość wielkości liter**
 - **master file table**
 - pliki w linuxie
 - zwykły plik, katalog, urządzenie, symbolic link
 - węzeł informacyjny (**inode**) + **blok danych**
 - ext2
 - brak problemu z fragmentacją
 - używany na USB
 - ext3
 - umożliwiające księgowanie
 - zapisywanie danych w dzienniku
 - w razie błędu brak utraty danych
 - tryby księgowania

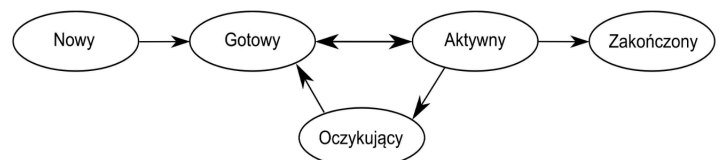


SO EGZAMIN NOTATKA

- dziennik - metadane i wartości pliku zapisywane w dzienniku
- uporządkowany - zapis w dzienniku tylko metadanych
- opóźniony zapis - okresowa aktualizacja dziennika, zapis tylko metadanych
- **ext4**
 - wydajność dla dużych plików
 - zmniejszona fragmentacja plików
 - stosowanie sum kontrolnych
 - szybkie sprawdzanie systemu plików za pomocą **e2fsck**
- podstawowe części extów
 - blok identyfikacyjny (superblock)
 - liczba i-węzłów, lokalizacja poniższych elementów itd.
 - tablica deskryptorów grupy bloków
 - lokalizacja pozostałych elementów
 - **mapa bitowa bloków**
 - **mapa bitowa i-węzłów**
 - dla map bitowych i-ty bit = 0: węzeł pusty, i-ty bit = 1: węzeł pełny
 - **tablica i-węzłów**
 - **bloki danych**
- i-węzły
 - coś w stylu stat()
 - poza tym bloki danych
 - **nie zawierają nazw plików**
- macierz dyskowa **RAID**
 - zbiór dysków
 - zwiększenie pojemności, odporności na błędy
- mbr - master boot record
 - program rozruchowy i tablica partycji
 - grub
- **grub**
 - opis wszystkich systemów operacyjnych
- mapowanie dysków do katalogów
 - /dev/hda1 -> /
 - /dev/hda2 -> /mnt/dysk1
 - /dev/hdb1 -> /mnt/dysk2
- fstab - informacje o systemie plików
- fdisk - tworzenie partycji itd.
- gparted

7. Procesy

- program - zbiór instrukcji
- **proces - część programu**
 - **wątek - w ramach jednego procesu, dzielona pamięć systemowa**
- program - statyczny
- proces - dynamiczny
- stany procesu
 - nowy
 - gotowy - procesor może wykonać instrukcje
 - aktywny - wykonywane instrukcje
 - oczekujący - czekanie na jakiś zasób (np. IO)



SO EGZAMIN NOTATKA

- gotowy i oczekujący może być wrzucony do swapa
- **lista procesów - windows**
 - task manager
 - tasklist
 - WMIC process
- **lista procesów - linux**
 - ps (ps ax) (ps -ef) (ps aux)
 - top
- **kończenie procesów - windows**
 - manager zadań
 - taskkill /F /IM firefox
 - taskkill /F /PID 2137
- **kończenie procesów - linux**
 - kill pid
 - kill -numer pid
 - killall nazwa
- **składowe procesu**
 - kod programu
 - zasoby pamięci, uchwyt
 - stan procesu
 - atrybuty
- **blok kontrolny**
 - po zmianie kontekstu dane programu lecą do bloku kontrolnego
 - i potem przy powrocie są przywracane
- ID procesu
 - PID - unikalny numer identyfikacyjny procesu
 - PPID - wartość pid procesu macierzystego
 - UID - user id
 - GID - group id
- **Fork**
 - tworzenie nowego procesu
 - tworzenie kopii procesu macierzystego
 - (Linux) proces o nazwie init o PID = 1 jest przodkiem wszystkich procesów
- **exec**
 - wymiana procesu na inny
- **exit**
 - zamknięcie wszystkiego
 - budzi rodzica
 - po exit proces staje się **zombie** i czeka aż parent odczyta kod wyjścia
- komunikacja procesów
 - pamięć współdzielona
 - pliki
 - sygnały
 - potok (np. pipe)
 - komunikaty
- obsługa wielu procesów
 - scheduler
 - kolejki zadań
 - przydzielanie czasu procesora
 - zmiana stanów procesów
 - rodzaje kolejek

SO EGZAMIN NOTATKA

- FIFO ~ FCFS
- SJF - shortest job first
- round-robin
- MLFQ
- fair-share

EKURS

1. Zaawansowane aspekty OS

- działanie programu (von Neumann)
 - pobranie instrukcji z pamięci
 - wykonanie jej
 - następna instrukcja
- zadania systemu operacyjnego
 - zarządzanie uruchamianiem procesów
 - jednoczesne działanie wielu programów
 - współdzielenie pamięci
 - dostęp do zasobów sprzętowych (abstrakcyjna warstwa dostępu)
- pamięć operacyjna
 - tablica bajtów
 - odczyt(adres)
 - zapis(adres, wartość)
- wirtualizacja procesora
 - iluzja nieograniczonej liczby procesorów
 - struktura do przechowania licznika, stosu, rejestrów
 - przełączanie między procesami
 - przy pomocy timera
 - IO
- time sharing
 - proces używa zasób przez krótki czas
 - system operacyjny przełącza procesy
- **składowe procesu 2**
 - instrukcje
 - przydzielona pamięć (np. stos, heap, itd) (czyli przestrzeń adresowa)
 - rejestry
 - licznik programu (PC)
 - wskaźnik stosu
 - lista urządzeń io
- **api do zarządzania procesami**
 - create - stworzenie procesu
 - destroy - usunięcie procesu
 - wait - wstrzymanie procesu
 - resume - wznowienie procesu
 - status - pobranie statusu
- **ładowanie procesów**
 - eager
 - ładowanie wszystkiego przed uruchomieniem
 - lazy
 - ładowanie danych i kodu w momencie potrzeby
 - stack(stos) - zmienne, parametry funkcji, adresy
 - heap(sterta) - dynamicznie alokowane dane

2. Multitasking

SO EGZAMIN NOTATKA

- **limited direct execution (ograniczone bezpośrednie wywołanie)**
 - dostęp do groźnych operacji pod kontrolą
 - odzyskiwanie kontroli nad CPU przez system operacyjny
- tryb pracy procesora
 - użytkownika
 - kod ma ograniczone możliwości
 - jądra
 - brak ograniczeń
 - tylko os działa w tym trybie
- **wywołania systemowe**
 - procesor uruchamia wywołanie systemowe, system operacyjny uruchamia kod do obsługi wywołania
 - instrukcja trap
 - skok do jądra
 - wykonanie wymaganej operacji
 - return from trap
 - co dzieje się ze stanami rejestrów procesora?
 - zależy od implementacji
 - mogą być przenoszone na stos jądra danego procesu (x86)
- wielozadaniowość bez wywłaszczania
 - OS ufa procesom
 - odzyskiwanie kontroli podczas wywołań systemowych
- **wielozadaniowość z wywłaszczaniem**
 - wsparcie sprzętowe
 - przerwania zegarowe
 - CPU wywołuje przerwanie
 - jest Timer, który przy starcie systemu jest zerowany
 - przerwanie zegarowe z określoną częstotliwością wykonuje wywołanie systemowe
- context switch
 - scheduler przełącza procesy na inne

3. Scheduler

- **FIFO** -> first in first out (**bez wywłaszczania**)
- **SJF** -> shortest job first (**bez wywłaszczania**)
- STCF -> shortest time to completion first
- planowanie rotacyjne (round robin)
- **MLFQ** (multi-level feedback queue)
 - kolejka priorytetowa
 - przydzielanie priorytetów
 - na podstawie zachowania
 - intensywne CPU - niski priorytet
 - IO - wysoki priorytet
- **priority boost - naprawia zagłodzenie**
- sprawiedliwe szeregowanie (np. losowe szeregowanie)
 - wyższy priorytet = większa ilość losów
- czas cyklu przetwarzania T_{cp} = $T_{zakończenia}$ - $T_{przedłożenia}$
- czas reakcji T_r = $T_{uruchomienia}$ - $T_{przedłożenia}$

4. Komunikacja w przetwarzaniu współbieżnym

- wątki = miniprocesy
 - działają w jednej przestrzeni adresowej
 - każdy ma własny licznik programu, rejestry, stos
- problem 5 filozofów

SO EGZAMIN NOTATKA

- 5 pałeczek
- żeby jeść trzeba mieć dwie pałeczki
- rozwiązania
 - naiwne:
 - weź lewą pałeczkę (czekaj na nią)
 - weź prawą pałeczkę (czekaj na nią)
 - jedz
 - odłóż pałeczki
 - problem: **zakleszczenie, zagłodzenie**
 - mniej naiwne
 - weź obie pałeczki (czekaj aż będą wolne)
 - jedz
 - odłóż obie pałeczki
 - problem: **możliwe zagłodzenie**
 - **hierarchia zasobów**
 - hierarchia pałeczek
 - weź pałeczkę o niższym numerze
 - jeden filozof ma obie wolne pałeczki i może jeść
 - problem: **niewydajność, tylko jeden filozof głównie je**
 - **rozwiązanie Chandy/Misra**
 - oznaczenie pałeczek - brudna, czysta
 - stan wszystkich pałeczek na początku = brudne
 - pałeczkę otrzymuje filozof o najmniejszym identyfikatorze
 - jeśli filozof chce jeść i nie ma pałeczki to wysyła do odpowiedniego sąsiada żądanie o pałeczkę
 - filozof, który dostał żądanie robi tak:
 - czystej pałeczki nie oddaje
 - brudną pałeczkę myje i przekazuje sąsiadowi
- **semafony** (zasoby międzythreadowe)
 - zasób krytyczny -> zasób, który może być używany w danym momencie tylko przez jeden wątek
 - przechowuje wartość
 - **sem_wait()**
 - zmniejsz wartość semafora o 1
 - jeśli wartość < 0: uśpij wątek
 - jeśli wartość >= 0: udziel dostępu do zasobu
 - **sem_post()**
 - zwiększ wartość semafora o 1
 - jeśli wątek czeka na zasób, to obudź go
 - chcąc pozyskać zasób sem_wait(), po skorzystaniu z zasobu sem_post()
- komunikacja międzyprocesowa
 - **sygnały**
 - przerwania programowe
 - obsługiwane przez jądro
 - **potoki (pipe)**
 - jak korzystanie ze zwykłych plików
 - zapis, odbiór danych
 - **jednokierunkowe**
 - **tylko między procesami ze wspólnej gałęzi**
 - **pliki fifo (named pipe)**
 - mkfifo

SO EGZAMIN NOTATKA

- **w dwie** strony
- pamięć dzielona (**Shared memory**)
 - współdzielenie pamięci między procesy (tak jak z własnej przestrzeni adresowej)
- gniazda(socket)
 - specjalne api - tworzenie punktów końcowych kanału komunikacji (jak plik fifo)
- kolejki komunikatów (message queue)
 - asynchroniczne

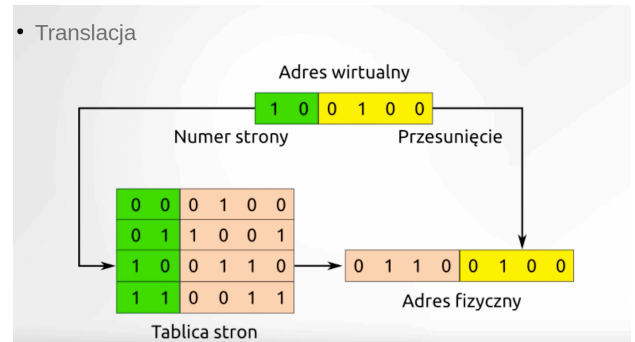
5. Zarządzanie pamięcią

- **przestrzeń adresowa**
 - fragment pamięci przydzielona procesowi
 - abstrakcja z punktu widzenia procesu
 - kod - kolejne instrukcje
 - stos - adresy powrotu, parametry wywołania, lokalne zmienne
 - sarta - dynamicznie alokowane zmienne
 - **sarta(heap) jest zaraz za kodem (rośnie w dół)**
 - **stos rośnie w górę (jest na dole przestrzeni adresowej)**
- wirtualizacja pamięci
 - adresy wirtualne na adresy fizyczne
 - wykorzystuje mechanizmy sprzętowe
 - podział na strony
 - **mmu - memory management unit**
 - położenie kodu: (void*) main
 - położenie sterty (void*) malloc(1))
 - położenie stosu (void*) &x
 - printf("%p", zmienna) adres pamięci
- u-obszar
 - znajduje się na samym początku przestrzeni adresowej
 - nawet przed kodem
 - proces nie ma do niego dostępu
 - malloc() i free() używają wielu wywołań systemowych
- translacja adresów jest wspierana sprzętowo
 - przesunięcie dynamiczne (**dynamic relocation**)
 - dla ciągłej przestrzeni adresowej o znanym, stałym rozmiarze
 - 2 rejestry - **baza, granica**
 - adres fizyczny = adres wirtualny + baza
 - granica sprawdza żeby proces nie uzyskał dostępu do pamięci poza przestrzenią adresową
 - segmentacja (**segmentation**)
 - przechowywana para rejestrów bazy i granicy na każdy segment
 - mogą być np 3 pary rejestrów - kod, stos, sarta
 - problem: **fragmentacja** -> dziury w pamięci
 - można kompaktować pamięć
 - można robić best fit - minimalizacja dziur
 - worst fit - zostawiaj największe wolne obszary pamięci
- **zadania systemu operacyjnego**
 - przydział pamięci (struktura free list ma mapę pamięci)
 - zwolnienie pamięci
 - podczas przełączania kontekstu system zapamiętuje bazę, granicę, fizyczne przeniesienie przestrzeni adresowej
 - obsługa wyjątków

6. stronicowanie

SO EGZAMIN NOTATKA

- podział pamięci na strony tej samej wielkości
- łatwe zarządzanie wolnymi Obszarami pamięci
- **tablica stron**
 - każdy proces ją ma
 - adresy translacji
 - **numer strony - n pierwszych bitów, przesunięcie - m następnych bitów**
 - **rozmiar strony = max offset**
 - eg. 110100, nr strony = 11, offset = 0100
 - tablice stron przechowują fizyczny adres strony
- co jeszcze przechowują tablice stron
 - bit poprawności - czy strona jest poprawna
 - bit ochrony - dostępność tej strony
 - **bit obecności - czy ta strona jest w fizycznej pamięci**
 - bit modyfikacji - czy była modyfikowana
 - bit użycia - czy strona była używana
- **operacje na stronach też wspiera mmu**
- pamięć podręczna translacji
 - ostatnio obliczone translacje
 - ASID address space identifier
- rozmiar tablicy stron
 - różne rozmiary stron
 - hybrydowe - stronicowanie z segmentacją
- **segmentacja ze stronicowaniem**
 - jedna tablica stron na segment logiczny (eg. jedna na kod, jedna na heap, jedna na stack)
 - rejestr bazy przechowuje adres fizyczny tablicy stron
 - rejestr granicy przechowuje rozmiar tablicy stron
- adresy w rozwiązaniu hybrydowym
 - numer segmentu + numer strony + numer przesunięcia
 - tablica segmentów -> tablica stron -> strona -> adres fizyczny = adres strony + przesunięcie
- wielopoziomowe tablice stron
 - podział tablicy stron na części - PTE
 - katalog stron - PDE
 - katalogi katalogów katalogów ...
- **swap space**
 - dysk twardy może przechowywać strony pamięci
 - o wiele mniejszy czas dostępu
 - pamięć operacyjna - ponad 2 gb /s
 - ssd - ponad 50 mb/s do 200 mb/s
 - hdd - ponad 10 mb/s do 100 mb/s
 - linux, windows - partycja swap
- page fault
 - wyjątek błędu strony
 - odwołanie do strony o bicie obecności = 0
- wymiana stron (swap)
 - fifo
 - lfu - least frequently used



SO EGZAMIN NOTATKA

- lru - least recently used
- **demand paging** - wymiana stron tylko przy odwołaniu, ładowanie używanej pamięci
- szamotanie - następuje ciągła wymiana stron - rozwiązanie - zamykanie procesów szamotających

INNE (opracowanie egzaminów)

- nazwy plików w unixowych systemach
 - zapisywane są w plikach będących katalogami
- **polecenia**
 - przepuszcza zadane pola i kolumny - cut
 - nazwa bieżącego katalogu - pwd
 - początkowe wiersze - head
 - jeden znak na inny - tr
 - usuwa plik - rm
 - pobiera opcje z linii poleceń - getopt
 - zmienia hasło użytkownika - passwd
 - utworzenie dowiązania twardego - ln
 - wysyła sygnał do procesu - kill
 - sprawdzić swój login - whoami
 - sprawdzić swoje id - id
 - sprawdzić aktywne procesy - ps
 - usunięcie katalogu - rmdir
 - utworzenie katalogu - mkdir
 - przeniesienie pliku do innego katalogu - mv
 - porównanie zawartości - diff, cmp, comm
 - sortowanie plików tekstowych - sort
 - poszukiwanie plików - find
 - edytowanie strumienia linia po linii - sed
 - zliczanie słów, znaków w pliku - wc
- **3 podstawowe zadania jądra systemu operacyjnego**
 - zarządzanie pamięcią
 - zarządzanie procesami
 - zarządzanie systemem plików
- **prawa dostępu w systemie linux**
 - jaka polityka dostępu do zasobów jest domyślnie stosowana w Linuksie?
 - Discretionary Access Control (Uznaniowa metoda kontroli dostępu)
 - kto decyduje o udostępnieniu zasobów?
 - właściciel pliku
 - jakie prawa ma uruchomiony proces?
 - prawa użytkownika
- **3 metody przydziału miejsca na nośniku danych**
 - Ciągła
 - Listowa
 - Indeksowa
- **nazwy metod obsługi urządzeń I/O**
 - procesor zleca urządzeniu wykonanie zapytania, a następnie okresowo je odpytuje czy jest gotowe - **polling**
 - po zleceniu wykonania zadania na urządzeniu procesor przełącza się na wykonywanie kolejnego procesu - **interrupt service routine (ISU)**
 - I/O ma bezpośredni dostęp do pamięci - **direct memory access (DMA)**
- **chmody poogarniać**
 - zmieniający prawa grupy i pozostałych tak, aby plik nie był dla nich wykonywalny

SO EGZAMIN NOTATKA

- `chmod go-x`
- nadający właścicielowi i grupie pełne prawa do pliku, pozostali bez zmian
 - `chmod ug=rwx`
- tak, żeby wszyscy mogli tylko czytać - `chmod 444`
- zmieniający prawa właściciela i grupy, tak żeby plik nie był dla nich wykonywable - `chmod ug-x`
- nadający grupie i pozostałym pełne prawa - `chmod go+rwx`
- wszystkim prawo do odczytu i zapisu - `chmod 666`
- właściciel może czytać i zapisywać, grupa może czytać i uruchamiać, inni mogą tylko czytać `chmod 654`
- pliku nie można uruchamiać nikt! - `chmod a-x`
- **stany procesu**
 - aktywny
 - gotowy
 - oczekujący
 - końcowy
 - początkowy
 - zombie
- **osoby**
 - Alan Turing - jeden z twórców informatyki, autor koncepcji maszyny Turinga
 - Richard Stallman - twórca GNU, założyciel FSF
 - Dennis Ritchie - twórca C, współtwórca unixa
 - George Boole - twórca algebry boola
 - Ken Thompson - współtwórca unixa, B i Go
 - Linus Torvalds - twórca Linuxa, gita
- **warunki polecenia test**
 - czy istnieje **-e**
 - czy istnieje i jest dowiązaniem symbolicznym **-L**
 - czy plik1 większy niż plik2 **-gt**
 - czy string1 jest różny od string2 **!=**
- **strumienie**
 - przekierowanie błędów jeden do dwa : jeden **2>** dwa
 - output jeden = input dwa: jeden **|** dwa
 - input 1 = dwa: jeden **<** dwa
 - output jeden dopisz do dwa : jeden **>>** dwa
 - znak pipeline to **|**
 - znak do uruchamiania procesu w tle to **&**
 - znaki grupujące polecenia tak, że nie powstaje nowy shell **{ }**
- **mechanizmy**
 - **umożliwiający asynchroniczną pracę urządzeń I/O**
 - buforowanie
 - umożliwiający działanie kolejki
 - priorytety
 - odpowiedzialny za obsługę pamięci
 - stronicowania
- **mechanizmy segmentacji na żądanie (?)**
 - wymiana stron następuje tylko w przypadku odwołania do strony
 - ładowanie tylko stron działających procesów
 - szybki start programów
- **5 zadań systemu operacyjnego**

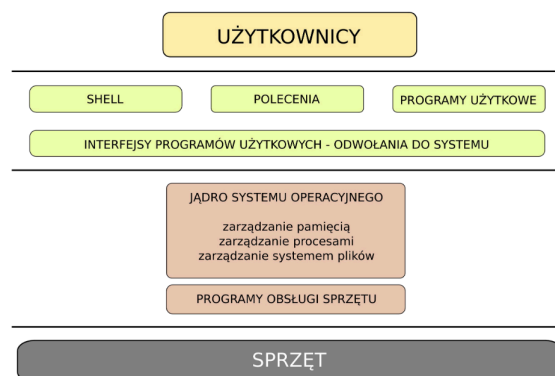
SO EGZAMIN NOTATKA

- planowanie i przydział czasu procesora
- kontrola i przydział pamięci operacyjnej
- dostarczanie mechanizmów synchronizacji zadań
- obsługa sprzętu
- obsługa wyjątków
- **3 modele kontroli dostępu do zasobów**
 - discretionary access control (**DAC**) - przydzielanie dostępu na podstawie własności (Linux)
 - mandatory access control (**MAC**) - administrator przydziela uprawnienia
 - role-based access control (**RBAC**) - uprawnienia przypisywane rolom
- **składowe trybu użytkownika w systemie Windows**
 - procesy systemowe
 - usługi systemowe
 - aplikacje użytkownika
 - podsystemy środowisko
 - podsystem dillek
- **składowe trybu jądra w systemie Windows**
 - warstwa wykonywalna
 - sterowniki
 - jądro
 - warstwa abstrakcji sprzętowej (HAL)
 - sterowniki i interfejsy graficzne
- **adres wirtualny -> adres fizyczny**
 - 4 pierwsze bity - numer strony, adres wirtualny = 11010101, tablica stron = {11001101, 11011111, 11100001, 11110110}, adres fizyczny = **11110101**
- **bity w tablicy stron**
 - bit poprawności
 - bit obecności
 - bit użycia
 - bit modyfikacji
 - bit ochrony
- **u-obszar**
 - lista otwartych plików
 - bieżący katalog
 - **tablica obsługi sygnałów**
 - parametry I/O
 - **proces nie ma dostępu do swojego u-obszaru**
- **i-węzeł**
 - rozmiar pliku
 - wskaźnik do bloków danych
 - liczba dowiązań
 - prawa dostępu
- **dane itd**
 - proces może przesłać, lub otrzymać dane za pośrednictwem **gniazda(socket)**
 - dostęp do zasobu może zostać wstrzymany za pomocą **semafora**
 - jedną z technik zarządzania pamięcią jest **stronicowanie** na żądanie
 - **fragmentacja** to zjawisko powstania wielu małych, niewykorzystywanych obszarów pamięci
- **sposoby minimalizacji fragmentacji**

SO EGZAMIN NOTATKA

- **best fit** -> dopasowywanie jak najlepsze(wolne obszary zbliżone rozmiarem) segmentów pamięci
- **worst fit** -> maksymalizacja rozmiaru wolnych bloków pamięci
- **kompaktowanie pamięci** -> okresowe układanie pamięci w ciągły obszar
- **układ sprzętowy odpowiedzialny za m. in. translację adresów = MMU (Memory management unit)**
- **metody wirtualizacji adresów**
 - przesunięcie dynamiczne, adres = baza + przesunięcie, granica
 - segmentacja
 - stronicowanie
- **systemy plików**
 - linux: ext4, ext3, ext2, ext, swap, btrfs
 - windows: ntfs, fat16, fat32
 - macos: hfs+
 - bsd: zfs
- **warstwowa architektura OS**
- **mechanizmy komunikacji między procesami**
 - potok
 - plik fifo
 - sygnały
 - kolejka komunikatów
 - pamięć dzielona
 - gniazda
- zadania typu uzupełnij kod

Warstwy systemu operacyjnego



ZAD 1 - przenoszenie plików pdf

```
find . -name "*.pdf" -exec mv {} /tmp
```

ZAD 2 - obsługa parametrów wywołania

```
if [ $# == 0 ]; then
    echo "Program musi zawierac parament"
elif [ $# == 1 ]; then
    echo "Program musi zawierać paramentr a lub b"
else
    while getopts ab: OPT; do
        case $OPT in
            a) echo "A" ;;
            b) echo "B" ;;
            *) echo "nieznany" ;;
        esac
    done
fi
```

ZAD 3 - wypisywanie liczb od 1 do 5

SO EGZAMIN NOTATKA

i=1

```
while [ $i -le 5 ]; do  
    echo $i  
    i=$((i+1))  
done
```

ZAD 4 - uzupełnij definicję funkcji

```
F () { wc nazwa_pliku | sort -n ; }
```

ZAD 5 - nazywanie plików mp3 nazwami typu 0001.mp3 według rozmiaru

nie znalazłem :(

