

Systemy komputerowe

Lista zadań nr A

Na zajęcia 25–28 maja 2020 (wersja robocza)

Należy przygotować się do zajęć czytając następujące rozdziały książek:

- Tanenbaum (wydanie czwarte): 1.1, 1.2, 1.4, 1.5, 2.1, 10.3, 11.4
- Arpaci-Dusseau: [Introduction](#)¹, [Processes](#)², [Process API](#)³, [Address Spaces](#)⁴

UWAGA! W trakcie prezentacji należy być gotowym do zdefiniowania pojęć oznaczonych **wytłuszczoną** czcionką.

Podręcznikiem do zadań praktycznych/prezentacyjnych jest „Advanced Programming in the UNIX Environment” (w skrócie APUE). Proszę przede wszystkim korzystać z podręcznika systemowego (polecenia man i apropos). Jeśli jego treść w *Linuksie* nie jest wystarczająco klarowna, to warto spojrzeć do odpowiednika z BSD – najlepiej z użyciem strony [mdoc.su](#).

W zadaniach oznaczonych literą **(P)**, wymagamy praktycznej **demonstracji** działania programów, najlepiej poprzez udostępnienie ekranu własnego komputera w usłudze Google Meet. By taka prezentacja była płynna należy przygotować w domu plik tekstowy z listą poleceń do wykonania i komentarzami. Każde zadanie należy mieć właściwie przygotowane do prezentacji przed zajęciami. W przypadku zbędnego przeciągania czasu odpowiedzi ze względu na problemy techniczne prowadzący ma prawo skreślić zadanie i postawić jeden punkt ujemny.

Zadanie 1. Wyjaśnij różnice między **powłoką** (ang. *shell*), **systemem operacyjnym** i **jądrem systemu operacyjnego** (ang. *kernel*). W tym celu dobrać kilka przykładów powszechnie wykorzystywanego oprogramowania. Jakie są główne zadania systemu operacyjnego z punktu widzenia programisty?

Zadanie 2. Czym jest **zadanie** w **systemach wsadowych**? Jaką rolę pełni **monitor**? Na czym polega **planowanie zadań**? Zapoznaj się z rozdziałem „System Supervisor” dokumentu [IBM 7090/7094 IBSYS Operating System](#)⁵. Wyjaśnij pobieżnie znaczenie poleceń **języka kontroli zadań** (ang. *Job Control Language*) użytych na rysunku 3 na stronie 13. Do jakich zastosowań używa się dziś systemów wsadowych?

Wskazówka: Bardzo popularnym systemem realizującym szeregowanie zadań wsadowych jest **SLURM**⁶.

Zadanie 3. Jaka była motywacja do wprowadzenia **wieloprogramowych** systemów wsadowych? W jaki sposób wieloprogramowe systemy wsadowe wyewoluowały w systemy z **podziałem czasu** (ang. *time-sharing*)? Podaj przykład historycznego systemu **interaktywnego**, który nie jest wieloprogramowy.

Ściągnij ze strony przedmiotu archiwum «prog1.tar.gz», następnie rozpakuj i skompiluj źródła poleceniem «make».

Zadanie 4 (P). Uruchom program «1_ls» pod kontrolą narzędzia «ltrace -S». Na podstawie śladu wykonania programu zidentyfikuj, które z **wywołań systemowych** są używane przez procedury: «`opendir`», «`readdir`», «`printf`» i «`closedir`». Do czego służy wywołanie systemowe «`brk`»? Używając debuggera «`gdb`» i polecenia «`catch syscall brk`» zidentyfikuj, która funkcja używa «`brk`».

¹<http://pages.cs.wisc.edu/~remzi/OSTEP/intro.pdf>

²<http://pages.cs.wisc.edu/~remzi/OSTEP/cpu-intro.pdf>

³<http://pages.cs.wisc.edu/~remzi/OSTEP/cpu-api.pdf>

⁴<http://pages.cs.wisc.edu/~remzi/OSTEP/vm-intro.pdf>

⁵http://bitsavers.org/pdf/ibm/7090/C28-6248-7_v13_IBSYS_Dec66.pdf

⁶<https://slurm.schedmd.com/SC17/SlurmOverviewSC17.pdf>

Zadanie 5 (P). Pod kontrolą narzędzia «strace» uruchom program «2_cat» korzystający bezpośrednio z wywołań systemowych do interakcji ze **standardowym wejściem i wyjściem**. Pokaż, że program oczekuje na odczyt na **deskrytorze pliku 0** i pisze do deskryptora 1. Naciśnij kombinację klawiszy «CTRL+D» kończąc wejściowy strumień danych – co zwróciło «read»? Zmodyfikuj program tak, by czytał z pliku podanego w linii poleceń. Co się stanie, jeśli przekazesz **ścieżkę** do katalogu zamiast do pliku regularnego?

Zadanie 6 (P). W systemach uniksowych wszystkie procesy są związane relacją **rodzic-dziecko**. Uruchom polecenie «ps -eo user,pid,ppid,pgid,tid,pri,stat,wchan,cmd». Na wydruku zidentyfikuj **identyfikator procesu**, **identyfikator grupy procesów**, **identyfikator rodzica** oraz **właściciela** procesu. Kto jest rodzicem procesu init? Wskaż, które z wyświetlonych zadań są **wątkami jądra**. Jakie jest znaczenie poszczególnych znaków w kolumnie STAT? Wyświetl drzewiastą reprezentację **hierarchii procesów** poleceniem pstree – które z zadań są wątkami?

Zadanie 7 (P). Znajdź pid procesu **X-serwera**⁷, a następnie używając polecenia «pmap» wyświetl zawartość jego przestrzeni adresowej. Zidentyfikuj w niej poszczególne **zasoby pamięciowe** – tj. stos, stertę, **segmenty programu**, **pamięć anonimową**, **pliki odwzorowane w pamięć**. Należy wyjaśnić znaczenie kolumn wydruku!

Zadanie 8 (P). Używając programu «lsof» wyświetl **zasoby plikopodobne** podpięte do procesu przeglądarki «firefox». Wyjaśnij znaczenie poszczególnych kolumn wykazu, po czym zidentyfikuj **pliki zwykłe**, **katalogi**, **urządzenia**, **gniazda** (sieciowe lub domeny uniksowej) i **potoki**. Przekieruj wyjście z programu «lsof», przed i po otwarciu wybranej strony, odpowiednio do plików «before» i «after». Czy poleceniem «diff -u before after» jesteś w stanie zidentyfikować nowo utworzone połączenia sieciowe?

⁷https://en.wikipedia.org/wiki/X_Window_System