1. Co to jest *race condition* (sytuacja wyścigu) w systemach współbieżnych. Podaj przykład występowania.
ODP:
Sytuacja, w której kilka procesów współbieżnie wykonuje działania na tych samych danych i w konsekwencji wynik tych działań zależy od porządku, w jakim one następowały nazywa się szkodliwą rywalizacją (lub sytuacją wyścigów)
Przykład: Dwa programy wykonują się równolegle. Pod wskaźnikiem mem znajduje się segment pamięci dzielonej do której piszą oba programy i nie zastosowano semaforów do ustalenia kolejności pisania

2. Wymień 5 cech procesu.
ODP:
- PID
- PPID
- UID
- informacje o otwartych plikach
- adresy i zawartość pamięci
- czas działania

3. Czy proces może przejąć obsługę wszystkich sygnałów? Jeśli nie to jakich nie może przejąć?
ODP: Nie, np.: SIGKILL, SIGSTOP

4. Proces stworzył potomka a następnie uległ zakończeniu. Co wskazuje PPID potomka?
ODP: Wskazuje na proces INIT (PPID=1)

5. Co to jest proces zombie i kiedy powstaje? Jak pisać programy aby nie tworzyć zombie?
ODP: W przypadku, gdy proces potomny kończy się w czasie, gdy jego proces rodzicielski nie wykonuje funkcji wait, wówczas kończący się proces jest umieszczany w stanie zawieszenia i staję się procesem zombie. Proces zombie zajmuje pozycję w tabeli utrzymywanej w jądrze dla kontroli procesów, ale nie używa żadnych innych zasobów jądra. Zostanie on zakończony, gdy jego proces rodzicielski zażąda potomka za pomocą wykonania funkcji wait. Aby nie tworzyć procesów zombie program rodzicielski powinien po wywołaniu potomka wykonać wait.

6. Co to jest efektywny UID procesu? Czy zawsze jest taki sam jak rzeczywisty UID?
ODP: Efektywny UID to identyfikator który wskazuje na użytkownika z którego uprawnieniami wykonuje się program. Efektywny UID jest inny, gdy plik wykonywalny miał ustawiony SUID i został uruchomiony przez innego użytkownika niż właściciel pliku.

7. Wymień 6 cech pliku w systemach unix. Jakie znasz funkcje systemowe pozwalające na wydobycie tych danych?
ODP:
Funkcje:
- stat
- fstat
- lstat

Cechy:
- prawa dostępu
- właściciel
- rozmiar
- czas modyfikacji
- typ pliku
- liczba dowiązań
- i-węzeł

8. Plik ma następujące prawa:
-rwsr-sr-x 1 philip staff 78837 Jul 25 17:36 /users/philip/plik
Po uruchomieniu go przez użytkownika jas z grupy studinfo jaki będzie:
(a) rzeczywisty UID procesu – ODP: identyfikator użytkownika jaś
(b) efektywny UID procesu – ODP: identyfikator użytkownika philip
(c) rzeczywisty GID procesu -– ODP: identyfikator grupy studinfo
(d) efektywny GID procesu – ODP: identyfikator grupy staff

9. Plik ma następujące prawa:
-rwsr-xr-x 1 philip staff 78837 Jul 25 17:36 /users/philip/plik
Po uruchomieniu go przez użytkownika jas z grupy studinfo jaki będzie:
(a) rzeczywisty UID procesu – ODP: identyfikator użytkownika jaś
(b) efektywny UID procesu – ODP: identyfikator użytkownika philip
(c) rzeczywisty GID procesu – ODP: identyfikator grupy studinfo
(d) efektywny GID procesu – ODP: identyfikator grupy studinfo
UWAGA!!! Wyjaśnienie praw dostępu znajduje się na końcu dokumentu

10. Jaki efekt powoduje tak zwany *sticky bit*?
ODP: Zabrania osobom innym niż właściciel i administrator usuwania i modyfikacji plików w katalogu z tym oznaczeniem. Natomiast każdy może tworzyć pliki w tym katalogu.

11. Co znajduje się w wirtualnym systemie plików /proc w systemach unix? Czy w linuxie występuje coś więcej?
ODP: Znajduje się w nim pseudosystem plików z informacjami o aktualnie działających procesach i systemie operacyjnym służący jako interfejs do struktur jądra systemu.
Dodatkowo w Linuxie możemy także przeczytać w tym katalogu o rzeczach nie związanych z procesami np. możemy poznać model procesora wpisując cat /proc/cpuinfo, w innych systemach to nie występuje.

12. Jakie polecenie systemu unix/linux pozwala poznać listę otwartych deskryptorów plików?
ODP: lsof

13. Jakie znasz funkcje pozwalające pobrać informacje o środowisku procesu?
ODP:
- getpid
- getppid
- getuid
- geteuid
- getgid
- getegid
- getsid
- getenv
- times
- getcwd

14. Jakie cechy procesu są dziedziczone po wywołaniu funkcji fork?
ODP:
- UID
- EUID
- GID
- EGID
- SID
- ustalenia dotyczące sygnałów
- bieżący katalog roboczy

Ogólnie: wszystko poza PID, PPID, oraz posiada własne kopie deskryptorów plików

15. W jakiej sytuacji wywołanie funkcji exec powróci do programu wywołującego?
ODP: W sytuacji błędu zostanie zwrócona wartość -1 do procesu wywołującego

16. Czym różni się funkcja execve od execle?
ODP:
execve- argumenty dla nowego procesu podawane są w postaci tablicy
execle - argumenty dla nowego procesu podawane są w postaci listy

17. Jakie cechy procesu potomka po wywołaniu fork różnią się od procesu przodka?
ODP: PID, PPID, oraz posiada własne kopie deskryptorów plików procesu macierzystego

18. Co robi funkcja ptrace?
ODP: Pozwala jednemu procesowi, śledzić i kontrolować wykonanie drugiego procesu

19. Podaj definicje semafora?
ODP: Semafor wykonuje dwie operacje atomowe: zamknięcie i otwarcie dostępu do zasobów

20. Dwa programy wykonują się równolegle. Pod wskaźnikiem mem znajduje się segment pamięci dzielonej (wspólnej):

program1:
```
int i=0;
if ((int*)mem[0]==0)
{
  (int*)mem[0]=1;
  for (i=1;i<10;i++) (int*)mem[i]=15;
  mem[0]=0;
}
```

program2:
```
int i=0;
if ((int*)mem[0]==0)
{
  (int*)mem[0]++;
  for (i=1;i<10;i++) (int*)mem[i]=0;
  mem[0]--;
}
```
Jakie niebezpieczeństwo może się pojawić przy wykonaniu tych programów? Jak można zaradzić podobnym sytuacjom?

ODP: Kolejność zapisywania danych jest nieprzewidywalna. Aby tego uniknąć należy zastosować semafory

21. Jakie cechy posiadają obiekty IPC w systemie V?
ODP:
- są tworzone w jądrze systemu i pozostają tam do czasu ich usunięcia
- nie posiadają nazw (nie są plikami)
- nie używają deskryptorów
- dają możliwość formatowania komunikatów
- dają możliwość pracy z komunikatami nie tylko w sposób FIFO
- Zapewniają możliwość kontroli przepływu komunkatów

22. Jakie znasz rodzaje obiektów IPC w systemie V?
ODP:
- kolejki komunikatów
- semafory
- pamięć współdzielona

23. Jakie znasz polecenia systemu unix pozwalające kontrolować obiekty IPC systemu V?
ODP:
- ipcrm
- ipcs

24. Jakie znasz polecenia systemu unix/linux pozwalające na podglądanie stanu i administrowanie procesami?
ODP:
- top
- kill
- job
- bg
- lsof
- ps
- fg

25. Jakie znasz funkcje do obsługi potoków?
ODP:
- popen, pipe, pclose – dla nienazwanych potoków
- mknod, open, write, read, close, unlink, – potoki nazwane (FIFO)

26. Czy dwa procesy mogą wykorzystywać jedna kolejkę fifo do komunikacji w obie strony?
ODP: Tak

27. W systemie plików jest nazwany potok fifo o nazwie ”kolejka”. Po wpisaniu następujących komend:
cat pewien_plik1 > kolejka&
cat pewien_plik2 > kolejka&
Co uzyskamy po wywołaniu
cat kolejka
Czy rozmiary plików pewien plik1 i pewien plik2 maja wpływ na to co uzyskamy na wyjściu?

ODP: Gdy plik *pewien_plik1* jest mały (mniejszy niż bufor *fifo*) dostaniemy na ekranie najpierw plik *pewien_plik1*, a następnie *pewien_plik2*. Natomiast gdy *pewien_plik1* jest większy niż bufor *fifo* to na ekranie zostaną wypisane przeplatane oba pliki.

28. Jak domyślnie zachowuje się funkcja read przy próbie odczytania potoku do którego nikt nie pisze?
ODP: Czeka, aż coś się pojawi do odczytania

29. Użytkownik wydziałowego systemu komputerowego posiada fifo o nazwie ”kolejka” w katalogu domowym i wykonuje następujące polecenia:
[philip@ultra60]$ cat plik > kolejka
oraz
[philip@aleks]$ cat kolejka
Co się stanie na skutek wywołania tych poleceń?
ODP: Na ekran zostanie wypisany plik o nazwie plik.

30. Użytkownik posiadający fifo o nazwie ”kolejka” wykonuje polecenia:
cat kolejka
oraz w innej sesji:
cat kolejka
oraz w jeszcze innej sesji:
cat pewien plik > kolejka
Jaki będzie skutek tych wywołań? Czy rozmiar pliku pewien plik ma wpływ na to jaki będzie rezultat?

ODP: Gdy *pewien plik* jest mniejszy niż rozmiar bufora, to w pierwszej sesji na ekranie zostanie wypisany cały *pewien plik*, a w drugiej nic. Natomiast gdy *pewien plik* jest większy niż bufor *kolejka* to *pewien plik* będzie rozdzielony między ekranami pierwszej, a drugiej sesji.

31. Dlaczego funkcja signal nie jest zalecana do obsługi sygnałów? Jakie funkcje są zalecane?
ODP: Ponieważ signal obsługuje sygnał tylko raz. Nawet, gdy od razu po wywołaniu obsługi przywróci sie ją, jest mały przedział czasu kiedy do procesu może przyjść sygnał, który będzie obsłużony domyślnie. Dlatego zalecana funkcją do obsługi sygnałów jest sigaction, która nigdy nie przestaje obsługiwać sygnału.

32. Czym różni się wywołanie systemowe stat od lstat?
ODP: Lstat jest identyczny ze stat(), lecz w przypadku gdy plik jest linkiem symbolicznym to zwracany jest status tego linku a nie pliku na który link wskazuje

33. Podaj przynajmniej jedno źródło informacji o użytkownikach w systemach unix. Jakie funkcje w C pozwalają te informacje pobrać?
ODP:
Źródła informacji: /etc/passwd /etc/group
Fucnkcje: - getgrgid - getpwuid

34. Podaj przykład funkcji pozwalającej wysłać sygnał do innego procesu? Jakie warunki muszą być spełnione żeby proces mógł wysłać sygnał do innego procesu?
ODP:
Przykład: funkcja „kill”
Warunki: proces musi mieć uprawnienia do wysłania sygnału, procesy muszą być na jednej maszynie, musi być znany pid procesu do którego ma być wysłany sygnał.
35. Jakie polecenie systemu unix/linux pozwala wysyłać sygnały? Podaj przykład wysyłając sygnał SIGUSR1 do procesu o PID 12345.
ODP: kill -USR1 12345

36. Do czego służą funkcje dup i dup2 ? Podaj scenariusz możliwego zastosowania.
ODP: Dup i dup2 powielają deskryptor pliku. Można to zastosować do przedefiniowania standardowego wejścia/ wyjścia/błędów

37. Czym różni się link symboliczny od dowiązania twardego ? Czy twarde dowiązania mają jakieś ograniczenia?
ODP:
link symboliczny: wskazuje on, odwołując się za pomocą nazwy, na dowolny inny plik lub katalog (który może nawet w danej chwili nie istnieć).
Dowiązanie twarde: jest to umieszczona w systemie plików referencja wskazująca na konkretny, istniejący wcześniej i-węzeł **w obrębie tej samej partycji**. Dla systemu operacyjnego, dowiązanie takie jest po prostu dodatkową nazwą dla wskazywanego obiektu - plik z n dowiązaniami ma n nazw.

38. Kiedy w systemie plików unix możemy mówić o usunięciu pliku? Co musi być spełnione aby miejsce zajmowane na dysku przez dany plik zostało zwolnione?
ODP: Muszą być usunięte wszystkie dowiązania twarde

39. Co zwraca funkcja time(time_t *t); pochodząca z nagłówka time.h?
ODP: Zwraca czas w sekundach od początku epoki tj. 01.01.1970 godz; 00.00

40. Podaj polecenia systemowe pozwalające zmienić właściciela/grupę/prawa pliku.
ODP:
- chmod – prawa do pliku
- chown - zmiana właściciela
- chgrp - zmiana grupy

41. Podaj przykład dwóch urządzeń znakowych w systemach unix (nazwa dowiązania w katalogu /dev)
ODP: /dev/random /dev/modem

42. Skąd pobierać liczby losowe w systemie unix? Czym dane te różnią się od tych generowanych przez funkcje rand() ?
ODP:Liczby losowe można pobierać z /dev/random
/dev/radnom generuje liczby losowe na podstawie różnych parametrów systemu i urządzeń które są nieprzewidywalne i zmieniają się w czasie. Natomiast funkcja rand() „losuje” liczby wg wzoru: $r_n=a*r_{n-1} \pmod b$

WYJASNIENIE PRAW DOSTEPU (wykonane przez: jarek24)

-rwsr-sr-x 1 philip staff 78837 Jul 25 17:36 /users/philip/plik

Jak odczytać te prawa?

Najpierw pogrupować po trzy omijając pierwszy myślnik

r	**w**	**s**
r	**-**	**s**
r	**-**	**x**

Teraz dodać wierszami wg przelicznika:
r = 4
w = 2
x = 1
- = 0

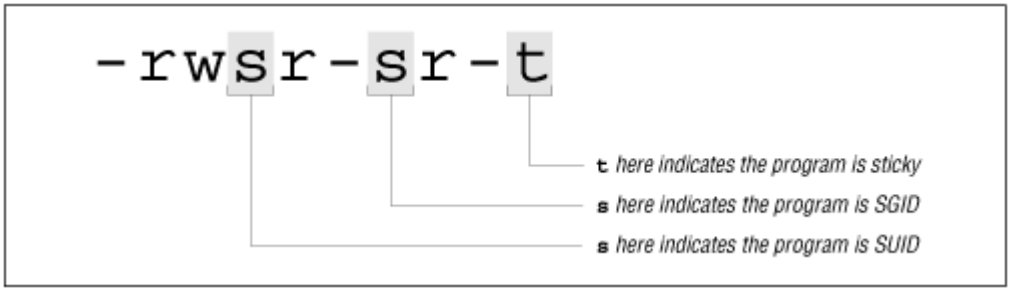
s = execute włączone więc dodać 1
S = brak execute więc dodać 0

Nasze polecenie chmod będzie wyglądało teraz tak:
chmod **abcd** plik

gdzie **a** odpowiada za bity specjalne, a **bcd** za standardowe uprawnienia (rwx)

Najpierw obliczmy wartość **b** = r+t+s = 4+2+1=7
Następnie **c** = r+s = 4+1=5
Następnie **d** = r+x = 4+1=5

Metoda obliczenia **a** będzie trochę inna bo tym razem interesuje nas tylko ostatnia kolumna która może zawierać takie wartości jak : - *s S x t* (S *s* i *t* mogą być tylko w wyznaczonych miejscach jak na rysunku):



Dodajemy wg przelicznika:

1 pole w kolumnie (wartość 4)
2 pole w kol. (wartość 2)
3 pole w kol. (wartość 1)

Jeśli pole z dowolnej kolumny jest jednym z następujących to:
s S t to dodajemy wartość jaką ma przypisane pole
w przeciwnym wypadku czyli gdy w kolumnie widnieje:
- *x* to dodajemy 0

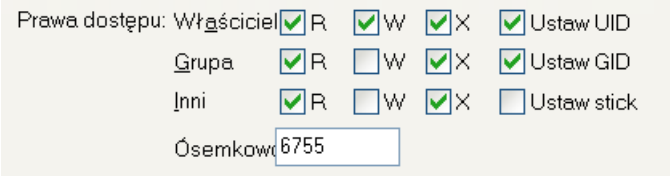
U nas sytuacja jest taka:

1 pole w kolumnie = s (tutaj jest s więc dodajemy wartość pola równą 4)
2 pole w kol. = s (tutaj też jest s więc znów dodajemy, tym razem 2)
3 pole w kol. = x (tutaj jest x czyli przeciwny przypadek, nic nie dodajemy)

a = s + s +x = 6 s+s != 2s

Więc zbierając to do kupy wpisując chmod **6**755 plik otrzymujemy: **-rwsr-sr-x**

Rozwiązanie graficzne:



To teraz wlasciwa czesc zadania:

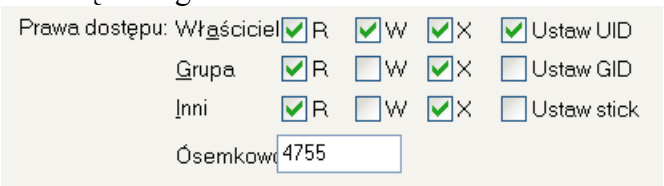
rwsr-sr-x 1 philip staff 78837 Jul 25 17:36 /users/philip/plik
Po uruchomieniu go przez użytkownika jas z grupy studinfo jaki będzie:

(a) rzeczywisty UID procesu = tego co odpala czyli jasia
(b) efektywny UID procesu = z racji bitu SUID jeśli Jas to odpali to jego proces otrzyma ID philipa
(c) rzeczywisty GID procesu = grupy Jasia czyli studinfo
(d) efektywny GID procesu = z racji bitu SGID otrzyma identyfikator grupy philipa czyli Staff

Kolejne zadanie już bez komentarza jak odczytać prawa:

-rwsr-xr-x 1 philip staff 78837 Jul 25 17:36 /users/philip/plik

Rozwiązanie graficzne:



Po uruchomieniu go przez użytkownika jas z grupy studinfo jaki będzie:
(a) rzeczywisty UID procesu = tego co odpala czyli jasia
(b) efektywny UID procesu = z racji bitu SUID jeśli Jas to odpali to jego proces otrzyma ID philipa
(c) rzeczywisty GID procesu = grupy Jasia czyli studinfo
(d) efektywny GID procesu = brak bitu SGID, więc także będzie studinfo, grupa Jasia